1995年11月1日発行（毎月1回1日発行）第14巻11号通巻163号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

特集 はりきって使うCD-ROM

データ解析の勘所/BMPファイルの活用/データトラックの読み出し方

S-OSディスクイメージファイルアーカイバSOSAR.X

新製品紹介 シャーペンワープロパックver.2.0/Xellent30PRO

11
1995



SOFT BANK オー／エックス
定価760円

SHARP

目の付けどころが、シャープでしょ。



■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page、LIPSIII、PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色(最大)のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

■「SX-WINDOW ver3.1システムキット」のお問い合わせは…シャープ(株)電子機器事業本部(液映)システム機器推進プロジェクトチーム 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ

フィールドが、膨らむ。

X68030
32bit PERSONAL WORKSTATION
&
X68000
PERSONAL WORKSTATION・XVI

68買ったら
EXEクラブ
へ入ろう!

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

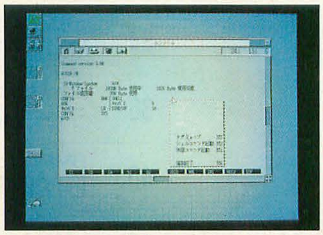
いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。



●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

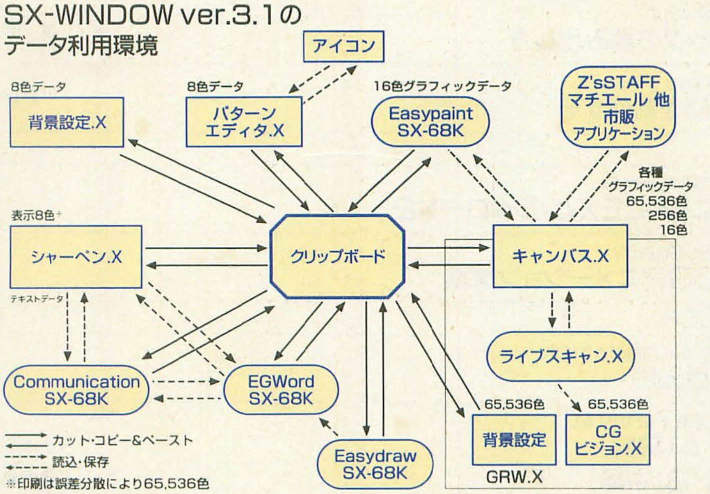


●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）



●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境



EXE
クラブって
何だ?

X68030/X68000を手に入れて、いろいろチャレンジしたい皆さん。情報のチャンネルは多いほどいいですよね。ということで、EXEクラブは68ユーザーのための水先案内人。あなたのチャレンジを強力にバックアップしますよ。

本体同梱の入会申込ハガキを送るだけで、自動的に無料入会。さらに下記の特典付き。

メリット 1
会員ナンバー入りのオリジナル会員電卓がもらえる。

メリット 2
各種フェアご優待・イベント案内等、数々の特典がある。

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD） 標準価格22,800円（税別）

シャープ株式会社

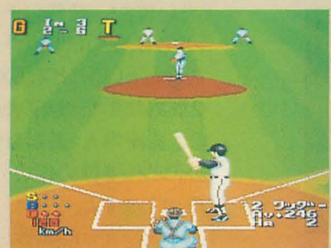
資料請求券
ペリフェラル
Oh!X
11係



特集　はりきって使うCD-ROM



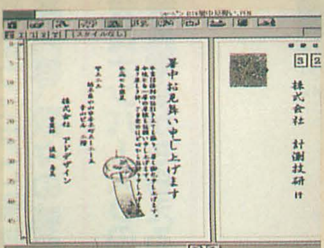
ナムコゲーム



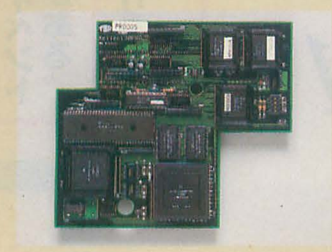
X68000ゲーム年代記



THE USER'S WORKS



シャーペンワープロパックver.2.0



Xellent30PRO

Oh!X

C　O　N　T

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　清瀬栄介　柴田　淳　瀧　康史　横内威至　進藤慶到　菊地　功　伊藤雅彦　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　加藤真二　●校正／グループごじら



表紙絵：塚田　哲也

1995 NOV. 11

E　N　T　S

■広告目次

ビデオグラフィックスの世界へ。



SHARP

目の付けどころが、シャープでしょ。

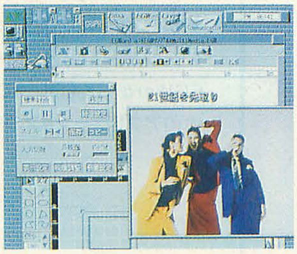
1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

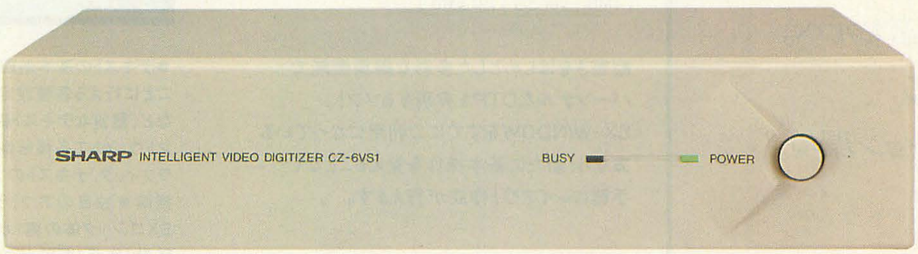
アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。



※SX-WINDOW版はバージョン3.0以降（メモリー4MB以上）、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上（メモリー8MB以上）が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。



■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2（FAST）インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2（FAST）対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、MacintoshIIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

for X68 Mac

ビデオ入力ユニット

CZ-6VS1

標準価格178,000円（税別）

SHARP

目の付けどころが、シャープでしょ

For X68030/X68000series

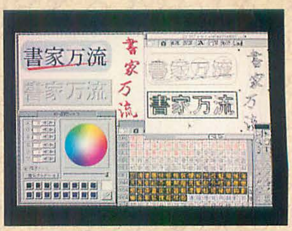
ORIGINAL SOFTWARE COLLECTION

さらに高度な創造次元へ。
ますます成熟する
そのアプリケーション環境。

アプリケーション

●独自のアウトラインフォントを付属

フォント&ロゴ デザインツール 書家万流 SX-68K



CZ-282BWD　標準価格29,800円（税別）

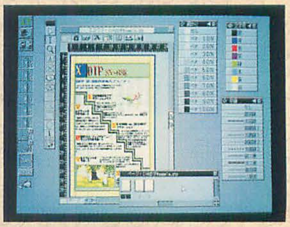
4MB | ver.3.0 | HD 10MB

フォントやロゴを手軽に作成するための
デザインツール。作成したロゴは
クリップボードを介し、シャーペンや
EGWord SX-68K、XDTP SX-68Kなど
他のアプリケーションで利用できます。

●SX明朝体/SXゴシック体フォント（JIS第1水準&第2水準）を付属●ベジェ曲線のアウトライン編集によるデータ作成●フォントファイル全体にわたってのエフェクト処理●既存のフォントファイルからのデータ抽出、ドローオブジェクトへのエフェクト処理●複数のフォントファイルをリンクして新たなフォントファイルの作成が可能●65,536色表示で確認しながらロゴ作成ができるグラフィックウィンドウ（GRW.X）対応

●パーソナルDTPをX68で

XDTP SX-68K



CZ-291BWD　標準価格35,000円（税別）

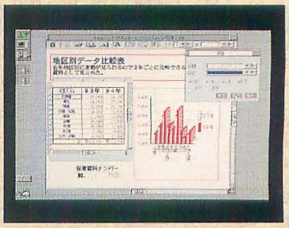
4MB | ver.3.0 | HD 5MB

縦書きをはじめとした多彩な編集機能で
パーソナルなDTPを実現するソフト。
SX-WINDOWをすでにご利用になっている
方なら、新たに基本操作を覚えることなく
手軽にレイアウト作成が行えます。

●テキストの基本処理をはじめ、テキストフレームごとに行える各種設定、スタイル別の検索/置換など、豊富なテキスト編集機能●グラフィックウィンドウ、そして各種画像フォーマットへの対応●グラフィック/テキストのフレームから独立した罫線機能●独自のアウトラインフォント（SX明朝体、SXゴシック体の第1水準）標準添付●ページの移動/作成/削除がスピーディに行える独立したページウィンドウをサポート

●DTP感覚で自在にレイアウト編集

Datacalc SX-68K



CZ-273BWD　標準価格59,800円（税別）

4MB | ver.3.0 | HD 3MB

SX-WINDOW対応の新世代統合ソフト。
表計算、グラフ、データベース、テキスト、罫線の
各データを1枚の用紙に重ね合わせ、
移動、サイズ変更など
DTP感覚でレイアウト編集ができます。

●カルクシートでは、セル番地を意識することのない直感的なセル指定が可能●データベースフィールドでは、同一項目でもデータ型/データ長の異なったデータを管理できるなど、自由な設計が特長●データベースフィールドで入力したデータをカルクシートのデータとして利用したり、カルクシートのデータ変更を自動的にグラフ表示に反映させたり、同一データからさまざまな分析が可能なデータリンクもサポート

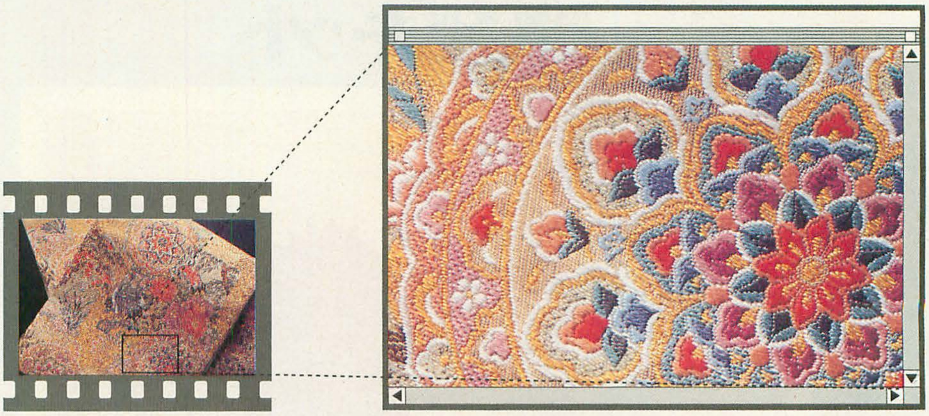
X68030
32bit PERSONAL WORKSTATION





高速・高画質、より深まる。
高速・高画質で人気のJX-330がさらに使いやすく! パワーユーザーも納得する実力を実現しました。
2400dpi※1
※1 2400dpiは当社独自手法による疑似解像度です。
＊イメージ写真です。
X68000対応カラーイメージスキャナ
JX-330X
SHARP
SHARP IS COLOR

SHARP

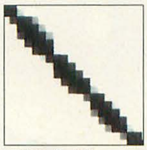


高スピード&高画質により、効率の良い作業を実現。拡大しても画像の荒れが少なく、レタッチ作業の短縮が図れます。
＊画面はハメコミ合成です。

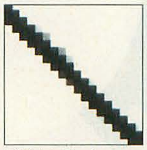
最高2400dpi※1の高解像度を達成。

基本600dpi、最高2400dpi※1の高解像度読み取りで、微細な線や点まで忠実に鮮明に再現します。縮小・拡大は30～2400dpiの範囲で設定可能です。また、約1677万色で原画に忠実なリアルな色合いを再現します。

●シャープ独自の「デジタルズーム機能」により、微細な線やズーム画像も忠実に再現。また、「ワンウェイスキャン方式」を採用し、凹凸のある原稿も鮮明に読み取りできます。



通常の拡大時
（当社従来機 JX-325）



デジタルズーム
（JX-330 シリーズ）



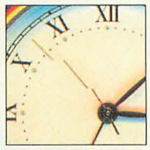
色の付いた影が出る
（当社従来機 JX-325）



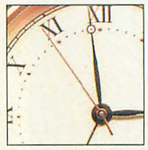
凹凸物も美しく再現
（JX-330 シリーズ）

クラス最速※2の高速読み取りを実現。

高速ヘッドリターン（約1秒）と高速読み取りを実現。A4、300dpiならカラー約13秒※3、モノクロなら約1秒※3で読み取りできます。最大A4/リーガルサイズ（216.4×355.6mm）までの原稿の読み取りが可能です。



読み取り速度16ms/
ラインのスキャナ



読み取り速度3.7ms※4/
ライン（JX-330 シリーズ）

透過原稿読み取りユニットとADFが同時装着可能。（オプション）

基本解像度600dpiまたは1200dpiの2種類の透過原稿読み取りユニットが選択使用できます。また、最大50枚までの同一サイズの原稿をスピーディーに自動送りできるADFも同時装着できます。



◀透過原稿読み取りユニット

▼ADF［原稿自動送り装置］



透過原稿読み取りユニット（オプション）	カラーイメージスキャナ	ADF［原稿自動送り装置］（オプション）
JX-3F6 標準価格 98,000円（税別）	JX-330X 標準価格 178,000円（税別）	JX-AF3 標準価格 58,000円（税別）
JX-3F12 標準価格 138,000円（税別）		

使いやすい高機能画像入力ソフトを標準装備 <JX-330X>　●Scanner Tool/s（画像入力ソフト）、対応フォーマット形式：ZIM,PIX,GL3,PIC,GLX,GLM

※1 2400dpiは当社独自手法による疑似解像度です。※2 クラスとは、A4フラットベットクラスのこと。'95年10月現在。※3 室温時（25℃）読み取り開始から読み取り終了までの動作時間。但し、初期動作及びデータ転送時間を除く。※4 室温25℃時。

■消費税及び配送・設置・付帯工事費・使用済み商品の引き取り費等は、標準価格には含まれておりません。
●ご使用の際は、必ず「取扱説明書」をよくお読みのうえ、正しくお使いください。●ご購入の際は、必ず「保証書」の必要事項をご確認のうえ、大切に保管ください。

■資料のご請求・お問い合わせは　シャープ株式会社 プリントシステム事業本部プリントシステム営業部
〒545　大阪市阿倍野区長池町22番22号　TEL.(06) 621-1221（大代表）　FAX.(06) 629-1207
〒162　東京都新宿区市谷八幡町8番地　TEL.(03) 3267-4410（ダイヤルイン）FAX.(03) 3260-2159
シャープ株式会社

SX-WINDOW ver.3.1 開発キット

著
◆
吉沢正敏
牛島健雄
西田文彦
小浜 純

B5変形判580ページ
定価5,800円
5"FD 1枚＋CD-ROM 1枚付



◆
本書の内容
◆

第1部
SX-WINDOW ver.3.1 開発入門

第1章
SX-WINDOWプログラミングの基礎

第2章
インストール

第3章
SX-WINDOW ver.3.1 開発キット

第4章
LIBSXC

APPENDIX
①
SX31KIT
②
LIBSXC便利帳
③
SX-WINDOW対応フリーソフト一覧

第2部
SXコール・リファレンス

本書は、シャープ提供の開発環境「Workroom SX-68K」と、
『追補版SX-WINDOWプログラミング』などで提供されたフリーソフトによる
開発環境を統合し、最新のSX-WINDOW ver.3.1の機能を利用した
アプリケーション開発環境を提供するものです。
添付FDには本書の著者たちが推奨する開発環境とCD-ROMドライバが、
添付CD-ROMには200本弱のSX-WINDOW対応フリーソフトを収録しています。
また、巻末にはver.3.1までのすべてのSXコールリファレンスをまとめてあります。

続刊

NetBSD/X68k

NetBSD/X68k委員会◆著
5"FD 1枚＋CD-ROM 1枚付き

SOFTBANK ソフトバンク株式会社出版事業部 〒103 東京都中央区日本橋浜町3-42-3 TEL03-5642-8101

SOFTBANK BOOKS

NOW
PRINTING

まぐろのすべて
MAGフォーマット誕生秘話

まぐろBBS 編著
CD-ROM付

パソコン通信の世界で標準的なグラフィックフォーマットであるMAGフォーマットの開発秘話。なぜMAGフォーマットを開発したのか、どのように開発を進めたのかといったところから、CGを描くノウハウに至るまで、さまざまな関係者達が自分たちの活動を明らかにしていく。付録CD-ROMには、ソースコード付きのMAG関連ツールのほか、マグロBBSなど活動しているメンバーによるCGを収録。

B5版・272ページ
定価 2,900円

CGネットワーカーズ自選作品集
33人のCG作家と8人のフリーソフト作者

PIRICA 監
Macintosh & Windows対応 CD-ROM付

本書には、パソコン通信や出版界で活躍中の33人のCG作家と8人のフリーソフト作者の作品を収録しています。本文中では、CG作家の自薦CGと作者紹介、ベテラン作家によるCGの描き方講座、各種CG関連のフリーソフトを掲載しました。また、新しい作品の入手方法と、CGをメインに活動中のBBS(パソコン通信)についても紹介します。付属のCD-ROMはWindows3.1とMacintoshの両方で使用できます。なお、収録したCGデータは多くの人に御鑑賞いただけるように圧縮や暗号化はおこなっていません。ローダーさえ用意すればMS-DOS環境下を含めて他の環境でもデータのみの利用が可能です。



B5変型版・160ページ
定価 2,900円

CGネットワーカーズ待望の第2弾
11月初旬発売予定!

定価は税込みです。
お近くの書店でお買い求めください。

ソフトバンク株式会社/出版事業部 SOFT BANK

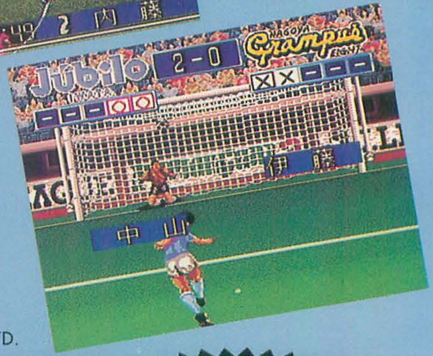
スーパーガイドシリーズ
話題のスーパーファミコンソフトを徹底攻略!!

キミもJリーグヒーローになれる！

Jリーグサッカー プライムゴール3 スーパーガイド



好評発売中



A5判・定価880円

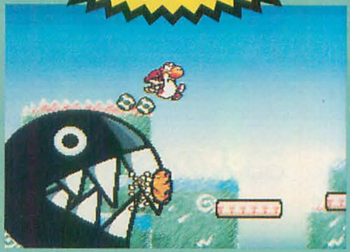
スーパーファミコン用の人気サッカーゲームシリーズ第3弾「Jリーグサッカー〜プライムゴール3」の完全攻略本。基本テクニックはもちろんのこと、今回から可能になったスルーパスを使った高等テクニックやゴールを決める必勝パターン解説、Jリーグ全14クラブの選手データやお勧めフォーメーション、おまけのミニゲーム攻略法などなど、プライムゴール3を楽しむための情報満載！

© 1995 NAMCO LTD.

ヨッシーアイランド スーパーガイド

好評発売中





A5判・定価880円

ヨッシーアイランドの完全攻略本。ヨッシーの基本操作から、各コースのポイント、さまざまなパワーアップアイテムなどをわかりやすく紹介。もちろん、全6ワールド、48コースの詳細なマップも見やすく掲載してあります。そのほか、ワールドごとに用意されているスペシャルステージやミニゲームの攻略法ももれなく解説！

これ1冊で、ヨッシーアイランドが120％楽しめる！

© 1995 Nintendo

第4次 スーパーロボット大戦 スーパーガイド

好評発売中



山猫有限会社 著 バンプレスト 監修

ガンダム、ゲッターロボ、マジンガーZなどの懐かしい巨大ロボットたちが暴れ回り、子供だけでなく20〜30代のファンを多く持つ「スーパーロボット大戦」シリーズ最新作の全てがわかる完全攻略本。シリーズ最多の全69マップ攻略はもちろん、272体にものぼるユニット（ロボット）および登場する223人全てのデータを収録。各ロボットの設定やシリーズの歴史、バックボーンの解説も付いている、まさに完全ガイド。

A5判・定価880円

■定価は税込みです　■お近くの書店でお求めください

ソフトバンク株式会社／出版事業部 SOFT BANK
販売局 TEL.03-5642-8101

［第54回］
袖摺り合うも68の縁

江口　響子

響子inCGわ〜るど

X68000ユーザーは少ない。

悲しいがこれは事実だ。持っていてももう使わなくなったという人もいる。実際に稼働しているマシンは日本全国でいったいどれぐらいだろうか。

でも，いいことだってあるのだ。

「おや，あなたもX68000ですか！」

「ええ……でも少ないんですよねえ，X68000持っている人は」

こんなふうに，お互い初対面でも会話はいとも簡単に始まる。少数派であるがゆえの連帯感だといってしまえばそれまでだが，これはこれでいいものだ。ユーザーなだけで知らない人同士が仲よくなれるマシンなんてほかにはない。DOS/VやPC-98，Macintoshでは，なかなか考えられないことだ。

「標準で6万色同時発色というのが嬉しいですよね。Macintoshでもエントリーマシンは3万色だから」

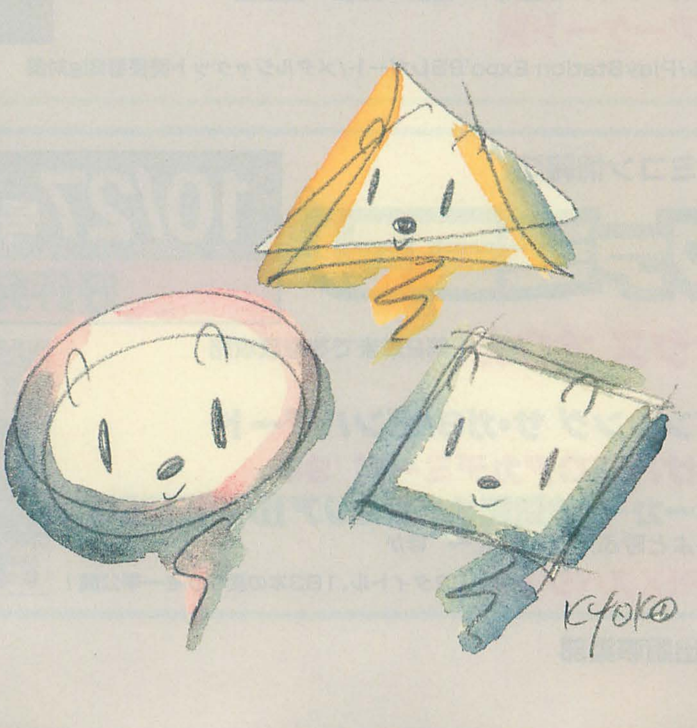
「現在，ゲームプログラマの中には，プログラムをX68000で覚えた人もずいぶんいるみたいですよ」

プログラムに詳しい人だともっと深い話ができるのかもしれない。自分自身，X68000でプログラムが組めないのをとても残念に思う。

私自身，X68000が縁で素敵な人たちと知り合うことができた。

本誌でお馴染みの，かまたさんをはじめとするDōGAの方々。純粋にコンピュータグラフィックの楽しさを追求してきたグループは，日本ではこのグループがピカイチだろう。

ひと昔前までは，CG制作はお金をかけなければできないものとされてきた。高価なワークステーションとソフト，それにメンテナンスの費用だってなかなかだ。初心者が始めるには，CGはあまりに高嶺の花だった。また，プロとしてCGを制作するようになると，時間をかけて楽しんで作ることが少なくなりがちだ。このことを，かまたさんはよく知っているのだと思う。だから，身近にあるX68000でアマチュアリズムにこだわりながら，素晴らしいDōGAのシステムを手数料だけで配布し続けているのだ。





また，X68000が好きだから，ぜひこのマシンで動かしてみたかったと，CG業界では有名なパーソナルリンクスのレンダラを移植してしまったIMAGICAの福本さん。「Photoshop」にもない機能をもっている「MATIER」を，ほとんどひとりで作り上げてしまった荒井さん。ほかのマシンとX68000のネットワークシステムを作った電机本舗の由井さん。X68000を使って4年越し一緒に仕事をしてきた小野さん。Oh!Xのライターである西川さんや丹さん，瀧さん，そして前田編集長をはじめとするOh!X編集部の方々……。

このほかにもまだまだたくさんいらっしゃるが，みな素敵な人たちばかりである（お会いしたことはないけれど，満開製作所の祝さんもたぶんそうなんじゃないかな，と思っている）。仕事だけでのつきあいだとなかなかこうはいかない。

もしかしたら，X68000を通じて得た一番の宝物はこの「人の縁」なのかもしれない。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
作成手順
使用ソフトはMATIER。大理石スキャン画像，スケッチ画像取り込み，ネガティブ，裏画面コピー，表画面コピーなどで画像処理。画像処理の部分だけRGB更新セーブ。



愛らしい猫娘たちの狂宴

戦え! こにゃんこEX

南国猫桜

本誌1993年8月号で紹介したカードバトルゲーム「ストにゃんダッシュ」を制作した南国猫桜が，今度は本当の格闘ゲームを制作してしまいました。それがこの「戦え！こにゃんこ」です。

ストーリーは，通っているみんながかわいいと評判の「私立猫耳女子学園」。そこでは不定期開催の「全学年統一異種格闘技祭」が行われており，今年は各学年から6人が参加することになった……う～む，実にわかりやすい。そして，基本的なルールは普通の格闘ゲームといっしょです。超必殺技（ねこみみ奥義）あり，バックダッシュあり，ふせありです。コマンド必殺技などの出やすさなどもちょうどいいくらいです。私が未熟なせいか，いわゆる「昇竜裂破コマンド」系といわれる，下方向に2回入る技をやろうとすると，ふせ（下をすばやく2回入れる）のあと，前ダッシュ（ふせの状態からボタンを押す）することが多くなってしまうんですけどね。このへんは慣れれば解決されると思うので問題ないでしょう。

操作はジョイスティック専用で，2ボタンが弱，中，強(両ボタン同時押し)攻撃に割り振られています。2ボタンだけだと，技の数が制限されるのでは？ と思うかもしれませんが，レバーを横に入れながらボタンを押すと違う技が出たりするので，見た目以上に技は多いようです。必殺技のコマンドは，どこかで見たことがあるようなものが多く，格闘ゲーム慣れしている人ならば，すぐにでも対戦できるでしょう。

対CPU戦は，難易度1のときでもそれなりに敵が強いので，格闘ゲームに慣れてない人にとっては，難しいかもしれません。グラフィックは綺麗だし，キャラはしゃべりまくるし，完成度はかなり高いソフトでしょう。

＜購入方法＞

郵便振替にて通信販売を行います。郵便局に備えつけの払込通知票で 2,500円(本体2,000円，送料500円)を振り込んで下さい。また通信欄には機種名，メディアサイズ，ソフト名「戦え！　こにゃんこEX」と「Oh!X係」を明記してください。

郵便振替口座：00120-5-609710　南国猫桜

＜問い合わせ先＞

〒183　東京都府中市武蔵台 3-38-34　西国ハイツ1F　南治方　南国猫桜











使えるキャラクターは全部で6人。スーパーゲージ，超必殺技やHIT COMBO(ゲームではCAT COMBOだけど)など，最近の格闘ゲームにある要素はひととおり網羅されています





ロード中に技の解説があるのは親切だね



とにかくニャーニャーよくしゃべるんだな

みんなで遊ぼうよ

キャラクターを選択して，対戦画面になるまでの間に必殺技一覧表が出ます。キャラクターを変えるたびに説明書を調べる手間が省けるので便利ですね。ねこみみ奥義は，強すぎず弱すぎずなので，これだけで試合の優劣は決まりません。使える回数も比較的多いので出し惜しみせずに使えます。でも状況を考えずに連発してもだめです。このへんのバランスのよさに好感がもてます。ハードディスクへのインストーラもついているので，より快適に遊べるし，キャラクターの声，グラフィックもばっちりです。雰囲気をよく出してます。ただストーリーがあるのに対CPU戦に勝っても，台詞もなくすぐ次のステージに進んでしまうところが味気ないかな。なんにしても重要な部分はしっかりしているので，仲間内で楽しむにはもってこいのソフトでしょう。ダルシムライクなキャラもいるのでダルシマーな方も安心ですよ。　（白井五三雄）

総合評価　★★★★★★★★☆☆



お宝目指してレッツラゴ〜！

シルクロード2

SPRITE



そんな便利な果物，僕もほしい



見てのとおりグラフィックは標準以上の出来。フィールドにいるかわいい敵を蹴散らしながら突き進もう。取得したアイテムをうまく使わないと進めない場所もあるのでちゃんと頭も使うように。要所要所に用意されたパズル要素も，ゲームを盛り上げてくれるぞ

ちょっぴりおちゃらけトップビュータイプのアクションゲームが，この「シルクロード2」です。

まず，起動して目に飛び込んでくるのがアニメーションをふんだんに使ったオープニング。この手のキャラクターものにはありがちだけど，非常に気合の入っている作りに好感がもてます。ゲーム中の操作は，トリガAで敵を攻撃できるシャボン玉を発射し，トリガBでアイテムを使うことができます（1画面中に1発で，4方向に攻撃可能）。アイテムは最初に持っているマップ以外にもいくつか存在し，マップ上に点在するボスを倒すことで入手することができるようになっています（選択しているアイテムは左上に表示されている）。

アクションゲームということで，シャボン玉をパシパシ撃ちながら敵を倒し，マップ上を探索するのですが，ときたま行けそうで行けないところやちょっとパズルめいた場所があります。このへんは頭とアイテムの使いどころ。うまく頭を使って切り抜けましょう。

そして，要所要所に体力回復アイテムが隠れているので（草むらを撃つと出てくる），ダメージを受けても安心です。しかもフィールドが切り替わると復活しているという親切設計（敵も復活するけど）がいいですね。だからといってゲームが簡単に終わってしまうかというとなかなかそうはいきません。フィールドに散りばめられた謎解きのほかにも，個性的なボスキャラたちが行く手を阻みます。結構凝った攻撃を仕掛けてくるので，倒すためにはそれなりに攻撃パターンを見切る必要があるので注意が必要です。無事，ボスを倒したあとには相棒（宝探しに巻き込んだ張本人ともいう）が現れ，敵を倒したときに出現するお金で買い物をすることができます。

誰にでも手が出せる範囲の難しさとかわいいキャラクター。バランスもよく，お手軽に遊びたい人にお勧めです。

〈購入方法〉

1,500円分（送料込み）の無記名定額小為替と住所，氏名を明記した宛名シール，そして希望するゲーム名とメディア（5インチ，3.5インチ）を明記したものを同封のうえ，下記の住所まで連絡すること。

〒558 大阪府大阪市住吉区我孫子1-2-18
松楓荘15号室　西村方　SPRITE



そんなこんなでお宝探しの始まりだ

適度な刺激が気持ちいい

適度に軽いゲーム全体のノリもいいし，作り自体もしっかりしています。ゲームのテンポを損なわない難易度の謎解きもグー。この謎解きのおかげで，敵を倒していくだけの単調になりがちなゲームが，とても楽しいものになっているといっていいでしょう。

問題があるとすれば，シャボン玉攻撃が単発で，なおかつ硬直時間があることと，アイテム選択をしている間でも敵は動いて攻撃してくることぐらいです（攻撃ボタンを押しっぱなしにすると，円状にアイテムが表示され，スティックの左右で選択できる）。この2点は，あらかじめわかっていないと結構つらいものがありますので，覚えておきましょう。

音楽もノリノリで画面の雰囲気に合っているしグラフィックもばっちり。あとは，キャラクターのアクションごとにPCMが鳴りまくれば，賑やかでさらに楽しくなったかもしれませんね。（浜崎正哉）

総合評価：★★★★★★★★☆☆

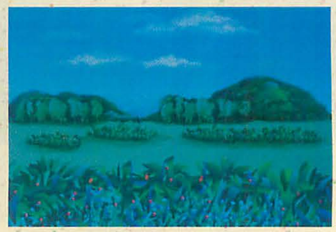
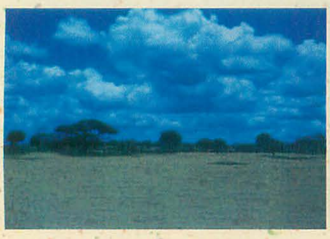
Oh!X Graphic Gallery

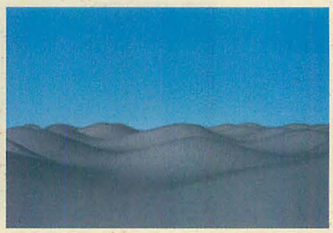
DōGA CGアニメーション講座

連載のコラムにありましたとおり，CGAマガジンの別冊「背景画像データ集」の一部をここで紹介します。内容は，手描きのものもあれば，写真をもとに合成したものもあり，ただの風景もあれば，心理描写用の背景もあります。
「背景画像データ集」の入手方法については連載のコラムを参照してください。

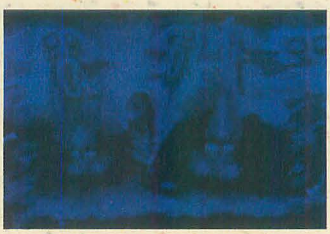
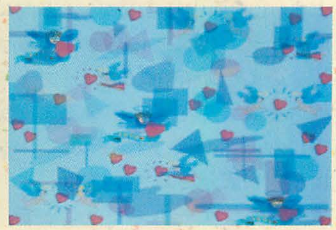


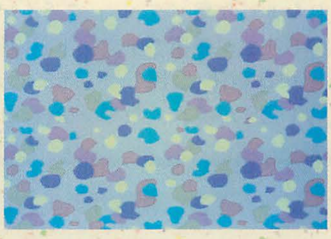
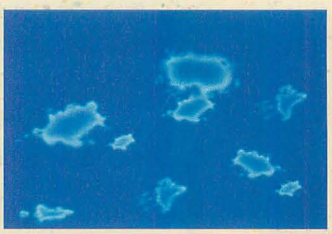






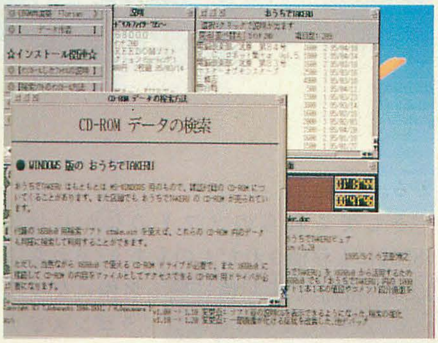







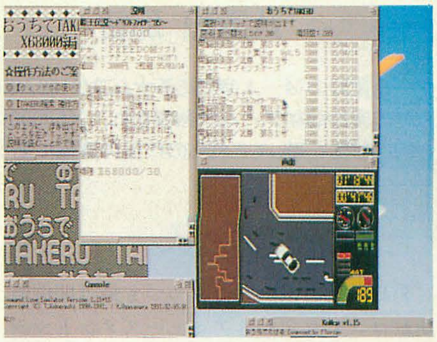
SOFTWARE INFORMATION

▶おうちでTAKERU X68000編



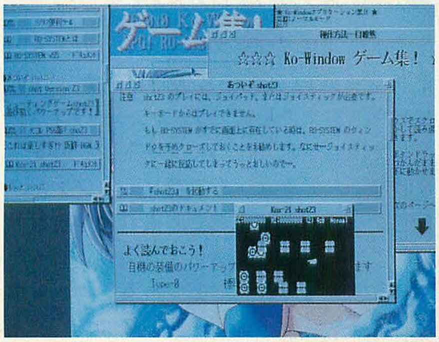
自宅にいながらTAKERUで販売中のX68000版のソフト情報を検索できる，カタログソフトが登場した。TAKERUで実際に使用している検索画面（静止画のみ）を，時間を気にせず1本1本じっくり見ることができるのは嬉しいかぎりだ。特にTAKERUを利用できない環境の人にとっては，貴重な情報源となるだろう。

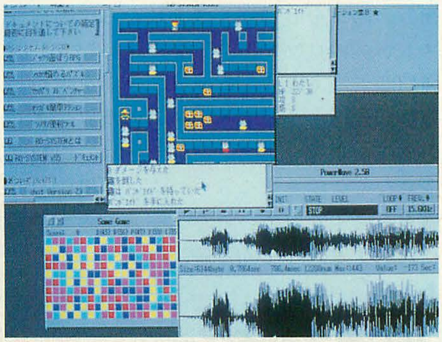
なお，このソフトを使用するためには，ハードディスクの空き容量が6Mバイト，メインメモリが2Mバイト必要になる。



X68000用　5″/3.5″2HD版　900円（税込）
TAKERU事務局　☎052(824)2493

▶Ko-Windowアプリケーション集8





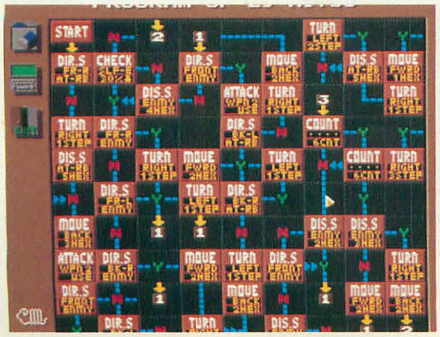
今回はゲーム特集ということで，さめがめ，シューティングゲーム，本ディスクの目玉といえるRPGシステムを使ったアプリケーションなどが多数収録されている。そのほかにも，PCMエディタやKo-Window用開発ライブラリなどのツールもあり，中身の充実度は保証付だ。

ディスクマガジンのような構成となっているため，面倒な操作は一切必要なし。内容がゲームだけあって，誰にでも気軽に楽しめるものとなっている。

X68000用　5″/3.5″2HD版　1,200円（税込）
TAKERU事務局　☎052(824)2493

▶R.C.ロボット集＋α Vol.6

自分でオリジナルロボットのプログラミングができることで人気を博した，対戦ロボットゲーム「ロボットコンストラクションR.C.」のデータ集「R.C.ロボット集＋α Vol.6」が発売になった（要「R.C.」本体）。今回は，第6回郵送ロボットバトル大会の結果とその記録，そしてユーザーの作ったフリーウェアが収録されている（ロボット管理，分析ツールやR.C.辞典など）。



また，ディスクには「R.C.」本体をバージョンアップするキットも含まれているので，古い「R.C.」本体プログラムに不満がある人や最新バージョンに触れてみたい人には，お買い得のソフトだろう。

X68000用　5″/3.5″2HD版　1,800円（税込）
TAKERU事務局　☎052(824)2493



発売中のソフト

★シャーペンワープロパック ver.2.0　計測技研　9/22
X68000用　5″/3.5″2HD版　9,800円（税込）
★おうちでTAKERU X68000編　TAKERU　9/30
X68000用　5″/3.5″2HD版　900円（税込）
★Ko-Windowアプリケーション集8　TAKERU　9/30
X68000用　5″/3.5″2HD版　1,200円（税込）
★R.C.ロボット集＋α Vol.6　TAKERU　9/28
X68000用　5″/3.5″2HD版　1,800円（税込）

新作情報

★EXCITINGみるく　TAKERU　12/未
X68000用　5″/3.5″2HD版　1,500円（税込）
★X CASE　Beシステム
X68000用　5″2HD版　19,800円（税込）
★Traum　象スタジオ
X68000用　5″2HD版　価格未定
★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定
★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円（税別）

ターゲットは87（前編）

Yaegaki Nachi 八重垣 那智



ゲーム業界の第一人者であるナムコが1987年に発表したシステムⅠ
数々の個性的なゲームが発表され，X68000にも移植が行われている
今回は移植を通してこれらの作品の真価に迫ってみよう

先日，秋の恒例となりつつも最近開催時期がコロコロ変わるアミューズメントマシンショウ，通称AMショウに行ってきた。昨年が「真侍魂」と「VF2」という大物で大騒ぎしていたのに比べると，今年はひたすら地味な機種ばかりでおとなしかった。その中で妙に注目されていたのは，ナムコが出していた，リバイバルゲームを3タイトル1セット＋アレンジバージョンの，6つのゲームが遊べるやつだ。ついつい遊んでしまったのだが，改めてオリジナルとアレンジを遊び比べると，オリジナルのよさがしみじみと感じられて，趣が深かったのは気のせいではないだろう。ナムコはプレイステーションでもリバイバルのセットモノを出していくとのことだが，よいものをいまに伝えること自体は間違っていないだろう。ただ，ちょっと後ろ向きではないかと心配してしまう今日この頃である。

振り返ればナムコ

ナムコはこの15年以上，常にビデオゲーム界のリーダーの1社として，その存在を誇示してきた。ナムコゲームの移植をレビューするたびに書いているが，ナムコには独創的なゲームばかり生み出していた黄金期ともいうべき80年代前半の時代があった。「インベーダーブーム」のあとに発売された，「パックマン」「ギャラガ」「ゼビウス」などに代表される一連のタイトルは，ビデオゲーム史上に残る作品ばかりだ。それをひとつのメーカーが独占して発表していたというのは，いくら黎明期であったビデオゲームの世界においても極めて特殊なことだったと考えてよいだろう。では，そのあとのナムコはどうなっていったのであろうか。

最近のナムコといえばポリゴンだ。しかし，これは「ウィニングラン」と「ギャラクシアン³」（花博版）以降の話であって，一応1990年以降の傾向である。これ以降の流れは記憶にも新しい人が多いだろうし，あまり行数も割かないが，ガンシューティングにレーシング，格闘，ポリゴンで固めた新製品からもわかるように，現在のナムコはポリゴンの時代だといっていいだろう。

すると，その間にあたる1980年代後半のナムコの特色はというと，それはファミコンとシステム基板の時代だということができる。どちらも考え方としては固定されたハードウェアで，ソフトウェアの内容と技術力の勝負をしていくことになる。開発コストが純粋にソフトウェアだけになり，開発期間も短くなるというメリットを生かし，発展期に入ったビデオゲームの世界に多くのゲームを送り出すということは，粗製濫造になる危険をはらみつつも，多様化して，飽きっぽくなっていったユーザーのニーズに対応する，現実的な解決方法だったのであろう。

繰り出されるゲーム

ナムコが最初にシステム基板によるビデオゲームを出したのは1986年のことである。このシステムはシステム86と呼ばれたが，発売されたタイトルは，1年ちょっとの間に6本であり，「リターン・オブ・イシター」や「源平討魔伝」という有名作を出しているものの，発売のペースとしては格段に速くなったという印象はなかった。

そして次に登場したのが，システム87と呼ばれる新型システム基板で，1987年4月のデビューから1991年12月まで，計21タイトルものソフトがリリースされた（表参照）。特に1987年は年末までに6本がリリースされているのだ。さらに同時に「ファイナルラップ」といったレースゲームなどもリリースされていたことを考えると，ソフト供給力のパワーアップは歴然としていた。途中でシステムⅠと改称したが，結果的にナムコ史上最も長寿命になり，ナムコを語るうえで無視することのできないタイトルがいくつも含まれている。この点で，システム87は，極めて重要な意味をもっているといえる。

そしてこれらのソフトのなかにはX68000にも移植されたものもあり，それらはX68000史上でも記憶に残る名作になっている。そのなかで今回は「ドラゴンスピリット」「ギャラガ'88」「メルヘンメイズ」の3本を取り上げ，移植を通してゲーム自体の存在価値に迫ってみたい。

初々しき青竜の伝説

まずは「ドラゴンスピリット」である。オリジナルのリリース当時は，ナムコのゼビウス以来の縦スクロールシューティングとして注目された，目玉タイトルである。ほどなく，ドラゴンを自機にしたファンタジックなコンセプトとその優れたサウンドが，ゲームファンを虜にすることになった。

8方向で竜を操り，2つあるショットボ

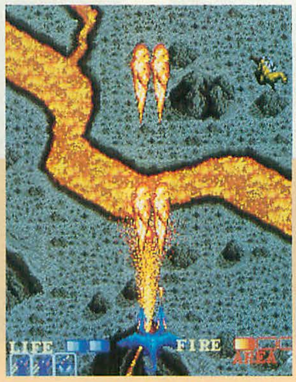
表　システム87（システムⅠ）の歴史

- 1987年
  - 1：妖怪道中記
  - [X]2：ドラゴンスピリット
  - 3：ブレイザー
  - 4：クエスター
  - [X]5：パックマニア
  - [X]6：ギャラガ'88
- 1988年
  - [X]7：ワールドスタジアム
  - 8：超絶倫人ベラボーマン
  - [X]9：メルヘンメイズ
  - 10：爆突機銃艇
  - [X]11：ワールドコート
  - 12：スプラッターハウス
  - 13：フェイスオフ
- 1989年
  - 14：ロンパーズ
  - 15：ブラストオフ
  - 16：ワールドスタジアム'89開幕版
  - 17：デンジャラスシード
- 1990年
  - 18：ワールドスタジアム'90激闘版
  - 19：ピストル大名の冒険
  - 20：倉庫番DX
- 1991年
  - 21：タンクフォース

注）[X]マークのあるものがX68000に移植されたもの。番号はリリース順に準拠



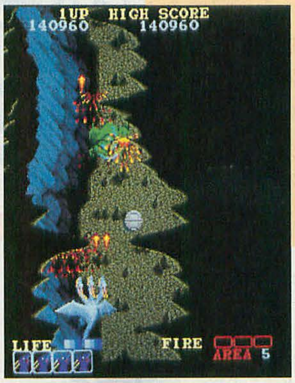
最初はのどかな風景で始まる



下レバーの裏技で噴火を避けろ



先手必勝。射たれる前に倒せ



壁が迫る中盤。操作は慎重に



北の大海原でコアラ(?)と対面



見えない位置の敵に要注意だ

タンで空中と地上の敵を炎を吐いて攻撃する。攻撃ボタンは独立しているが，竜の首の都合(笑)で同時には攻撃できないのが特徴だ。さらに困ったときに発射するボンバーや溜め射ちなどという代物もない。きわめてシンプルなゲームだ。同時期にリリースされた縦スクロールシューティングとして「飛翔鮫」と「1943」を挙げることができるが，これらがステージイメージを全体で統一しているのに比べ，ドラゴンスピリットはゲーム展開の多彩さで勝負しており，際立った違いを見せている。火山や砂漠といったものから水中に暗闇まで，アイデア目白押しのステージがゲームに用意されているのだ。

旧来のナムコが，目先を変えることよりも数少ないパーツを組み合わせることでゲームの奥深さを演出していたことを考えると，これは大いなるターニングポイントになったというにふさわしいゲームである。こういったバラエティに富んだ内容に，さらわれた姫を助けにいくため竜に姿を変えた主人公という設定などが，作り込みの細かさを支えているのである。

この人気作になった「ドラゴンスピリット」のX68000版は，発売前から大きな話題を呼び，オリジナルの興奮覚めやらない1988年末にリリースされている。移植の担当は電波新聞社だ。移植された当時，ナムコの直営店なんかを覗くと「ドラゴンスピリット」はチラホラ残っていたため，もうそんなゲームがパソコンに移植されているのかとX68000ユーザーになっていなかった私などは，舌を巻いたものだ。

天を駆け闇を裂け

実際の移植自体は，いま見てもほぼ申しぶんのない出来である。画面を横にするモードがついており，モニタを立てると本物そっくりの感覚で遊べるというのが，セールスポイントのひとつだったのは，ある意味では驚くべきことであろう。モニタが壊れる危険を知りつつも，ついつい縦にしてしまった人も少なくないと聞いている。

ほかにもオリジナルにあった2つのバージョンを設定で切り替えてプレイできたり，直前のプレイをトレース再生することができるという機能など，凝りに凝っている。ただ，エンディングのBGMがサウンドテストで聞けないところまで似せる必要はないような気もするし，バージョンとライフゲージは別個に設定できるようにするべきだ，というような注文もあるが，ここまでこだわるといきすぎになるかもしれない。

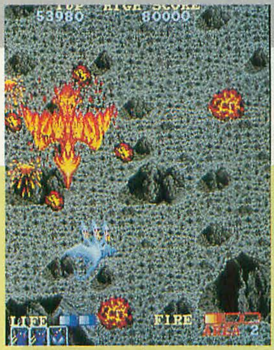
とまあ，手放しで褒めているが，実際のところ，この「ドラゴンスピリット」は，かなり記憶系に偏ったパターンシューティングである。1面はともかく，バラエティに富んだステージのモチーフに沿った仕掛けが目白押し。一部のボスは，最初の1，2秒の動きで勝負が決まるし，知らないだけで即死する場所だって少なくない。継続プレイも7面以降に進んでも7面からしかできない。

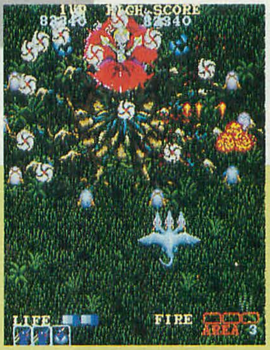
そういう点を見ると，さすがに8年前のゲームという印象は否めない。いまから入門するとしたら，敷居はかなり高いといえるだろう。しかしX68000にタイムリーにそっくりなアーケードゲームを移植するという，最もインパクトのあるスタイルの成功例として絶対に挙げられるこの作品は，もちろんユーザー必修の逸品である。

迫りくる宇宙の虫

「ドラゴンスピリット」の半年後には，1987年に出たのに'88とついている，名作「ギャ











待ち構える数々のボスたち。どれもあらかじめ作戦を立てて一気に始末しないと，とんでもない目にあうことも。弱そうに見えても油断は禁物だ。



最初は吸われてトリプルを狙おう



やはりパーフェクトは気持ちがいい



自信がない場合は黙って眺めよう

ラガ」の続編，「ギャラガ'88」がリリースされている。前作がシンプルで誰にでも遊べる内容で超ロングヒットしただけに，これも期待が大きかった作品だといえるだろう。正確には「ギャラクシアン」→「ギャラガ」→「ギャプラス」→「ギャラガ'88」なのだが，あえて「ギャラガ」という名前を使ったところに，このゲームに対する期待と意気込みを感じることができる。

このゲームのルールは単純であり，画面下段を左右に動いて，ショットボタン1個で飛び回る敵を撃ち落とせばよい。「ギャラガ」はこれに，自機を敵にいったん奪わせておき，あとで取り返すことで2つの自機を並べて倍の威力で攻撃できるという，デュアルファイターというシステムが採用されていた。もちろん「ギャラガ'88」でもこのシステムは健在で，なおかつパワーアップされている。それは，デュアルファイターを敵に奪わせることでトリプルファイター（同名のヒーロー特撮とはまったく関係ない）まで成長させることが可能になっているのだ。しかし，パワーアップをしたぶんだけ，しっかり強い敵が出てくるようになるので，決して楽になってはいない。

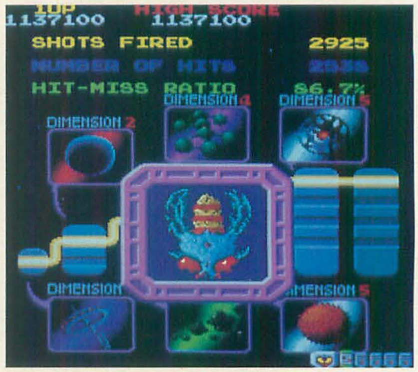
ほかにもギャラガで好評だったチャレンジングステージも健在だし，ゲーム中に登場するカプセルを回収することで，より高度で強い敵の登場する次元にワープすることもできる。次元は，スコアやエンディングの善し悪しに関わってくるので，できるだけ上げたほうがいい。

しかし，高次元では敵全体がパワーアップし，特定のタイミングでないと倒せない敵などが出てくるので苦戦は必至となる。自分の腕前と相談して調整するほうがいいだろう。

目指せ宇宙のヒーロー

「ギャラガ'88」は「ドラゴンスピリット」と同様に，オリジナルは画面を縦向きに使用している。そのため移植版では，横につぶれたような画面構成が基本となっている。画面幅を狭くしたモードもあるが，「ドラゴンスピリット」のようにモニタを縦置きさせるような機能はついていない。

そういう見方をすると，機能的な見栄えは悪いのだが，ゲームの内容の単純さを考えると，この場合はそれほどゲーム性を損ねていないことがわかる。実際遊んでみても違和感は少なく，手軽に遊べる作品としての魅力があるのは，オリジナルからの移植がしっかりしているからだといっていいだろう。



最後のボスの魚拓ならぬ虫拓を入手

そして，移植の担当は「ドラゴンスピリット」と同じ電波新聞社だが，その頃の同社の移植もののセールスポイントになっていたアレンジ要素というのが，この「ギャラガ'88」にもついている。といってもチャレンジングステージのキャラクターとパターンの差し替えという，結構地味なアレンジなのだが。まあ，ほかのナムコゲームのキャラクターを流用し，音楽もそういったものを基本にアレンジしているので，ナムコファンには違和感はないだろう。名前どおりギャラクティックダンシングとして，鑑賞するのがスジであろう。

とにかくゲームがシンプルなだけに，飽きがくるのが早いかもしれないが，逆に思い出したとき，暇なときに気楽に相手をしてやるゲームだと考えることもできる。特にゲーム慣れしていない人に遊ばせてあげるのにピッタリなのが，「ギャラガ'88」なのである。

夢の空中迷路

最後に紹介するのは「メルヘンメイズ」である。このゲームは「ギャラガ'88」の半年後にリリースされた，

計画的にカプセルを回収してディメンジョンを上げていくのがハイスコアへの近道になるぞ







ディメンジョンが最高だと，最終面で敵の本拠地に突入し，真のボスとの最終決戦が……









鏡を潜りぬけて世界を渡り歩こう



世界を移動中。えっくすの国はない



動く床，迫る敵。あせりは禁物だ

システム87(当時はもうシステムIと呼ばれていた)に最も脂が乗っていた頃の作品である。

このゲームで最も特徴的なのは，このゲームがナムコオリジナルではなく，とある別のメーカーによる作品であるという点である。このことは一般にはほとんど公表されていないので，いわれてみるとサウンドやキャラクターの雰囲気が，当時のほかのナムコゲームとは違うように感じられるかもしれない。しかし，その内容は，往年のナムコをほうふつさせるようなアイデア重視のゲームとして，高い完成度を誇っているのである。

ゲーム内容は8方向移動によって，クオータービューの足場状のステージを進み，シャボン玉のショットで敵を倒していくのが基本だ。レバー移動では足場から落ちることはないが，敵の攻撃(弾や体当たり)を食らって弾き飛ばされたときに足場から押し出されると，そこで初めてミスになるという特殊なシステムとなっている。

また，離れた足場を渡るときや敵の攻撃を避けるときはボタンでジャンプできるのだが，着地の位置に足場がないとそのまま落下して，これもミスになってしまう。こうしたなかで障害物をかわし，動く床に飛び乗り，ボスキャラを倒していくのである。

ウサギのいうがままに戦う

この「メルヘンメイズ」では，舞台が鏡の中の不思議な世界ということで，各ステージには，お菓子や機械に虫といったモチーフが与えられている。が，実際にプレイしてみると仕掛けは基本的なパーツや足場の形状の組み合わせであり，画面の雰囲気こそ違ってもやることは一緒であることがわかる。その点では，やっぱりナムコっぽい緻密さが感じられるといえるだろう。

ただし，実際に遊んでみると画面のキャラクターの見た目の可愛さの何倍もハードなゲームであり，アクション性の高いシビアなゲームであることに気づく。一応，可愛いキャラクターに魅力を感じる，ふだんゲームをやらないような女の子，といった層も意識しているはずなのだから，もっと難易度を低くしてもよかったのではないかという気がする。

そんな見かけと裏腹の「メルヘンメイズ」であるが，X68000の移植としてはオリジナルの厳しい部分をしっかりと継承した仕上がりとなっている。むしろ移植するにあたって，なにか簡単にする工夫がほしかったような気がするぐらいである。そういったアレンジ的な要素やサービスモードがないのは，この移植を担当したのがSPSであるということとも無関係ではないだろう。

ただ，昨今ではこういったジャンプのアクション性を意識した緻密なゲームというのは絶滅してしまった感があり，そういった貴重なジャンルの逸品としての存在価値は大きい。正確な操作とタイミングを見切る判断力を刺激する意味で，食わず嫌いの人に意外とお勧めである。

ゲームは時代を映す

今回は3本を取り上げてみたが，実際にこれらのオリジナルを出していた当時のナムコは，正直いって，一般受けするようなゲームを出せずに苦しんでいるように見えていた。風営法が施行され，ゲームがアイデアだけではない時代になり，そのなかでナムコは従来のビデオゲームの成果とのギャップに悩んでいたのだろうと思っている。プレイヤーの層が徐々に拡大し，その趣味嗜向が分散していく中で試行錯誤していた多くのゲームは，いろいろなことを教えてくれるはずである。次回は残りの3つを取り上げて考察してみたいと思う。



間抜けだけどボスはボス。倒せ！



無敵のうさぎさんバリア。突進OKだ



時間と締切は，守らないと大変なことに



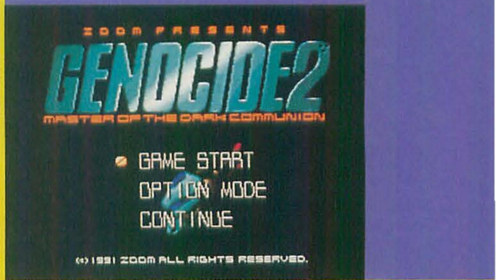
油断すること，すなわち落ちること。でも風船があれば1回だけ落ちても大丈夫

秒刻みの華麗なる戦闘(中編)

Yokouchi Takeshi 横内 威至

G2タイムアタックへの道も中盤戦に突入だ
さらなる厳しい戦いが待ち受けているが，ひるまず立ち向かおう
集中力をぎりぎりまで高め，限界に挑戦するのだ

●ジェノサイド2



ズーム　☎011(613)0191

STAGE 3

3-1 目標：55秒

いよいよ，ステージ3-1。ここで初めてステージ開始直後に現れる体力回復アイテムを使う。体力を補充している時間は限りなくムダだ，といえるくらいまでここまでのプレイがほぼ完璧にできるようになろう。

さて，このステージもステージ2-3同様にベティーの余裕がわずかしかない。おそらくベティーの残り時間は1秒未満となるだろう。特に前半が重要な部分なので，マップを見ながら攻略に励んでもらいたい。

図のタイミングで飛び，ベティーを溜め，敵を破壊する。まず，③の丸くて水色の奴はMAXベティー1発で仕留めなければならない。これはベティーを投げずに敵に当てればOKだ。破壊後はすぐにエネルギーを溜めること。そして，④，⑥も同様にして始末し，ベティーのエネルギーを溜めながら右へダッシュ。画面奥に⑦が見えるがとりあえず無視し，ベティーのエネルギーがMAXになると同時に右からくる⑧をまず殺す。そのあと，ベティーのエネルギーを溜め続け，自分とX座標の位置が合うと飛んでくる⑦にMAXベティーが当たるように位置を調整する（判定も飛んできた瞬間からなのでこのタイミングをつかんでおくこと)。ここはレバー上でやや上方にベティーをずらしておくのが吉。そのまま溜めていれば，最後の奴⑨もベティーのエネルギーがMAXになった瞬間にピッタリ判定がつくので簡単に片づけられる。そして0.5秒後にベティーが消え去ることになるだろう。



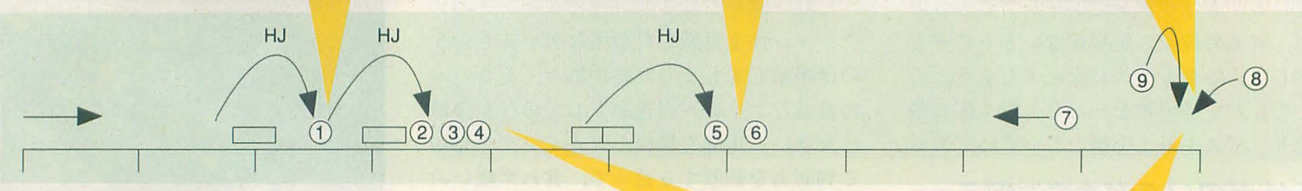
▶①の戦闘機をハイジャンプで越え、着地と同時にベティーを用意。この時点からベティーのエネルギーを溜め始めること



▶⑤の戦闘機をハイジャンプで飛び越え、⑥を破壊しよう。位置合わせを慎重にしないと⑥に突っ込むことになるので注意



▶⑦を無視して走り抜けると、ちょうどベティーのエネルギーが、MAXになったところで⑧がベティーに突っ込んでくる



▲図中の記号は，HJがハイジャンプ，□が画面上に固定されている敵戦闘機だ。①，②，⑤の移動する戦闘機は破壊することも可能だが，倒しても無意味なのでさっさと飛び越えてかわそう。ジャンプ中は高速移動を使い，さらに位置合わせも正確に行おう。ここはかなりの難易度なので，ジャンプする場所，当てるタイミング，敵の位置すべてを脳髄に刻まねばならない。うまくいけば全面中，1，2を争うほどスピーディでイカしたプレイが可能だ。ぜひともG2の置いてあるバーで披露して女を口説きたいね



▶③にMAXベティーを当てて破壊し、④に備えていったん引き返しベティーのエネルギーを溜めろ



▶画面奥から迫ってくる⑨も、タイミングさえ合えば、勝手にMAXベティーに突っ込んでくれる。楽勝だぜ

▶後半はミサイル発射口を高速3段斬りを使いつつ破壊し，艦橋の上にあるEXPLODEを奪う。そして，丸くて水色の奴を飛び越えてクリアだ。カッチョいい飛行機で空へと飛び立ち，次のステージへ









3-2 目標：50秒

ここは，結構いい加減にプレイできるところだ。

まずは鈍臭い奴とバズーカ男が登場するので，鈍臭い奴だけにターゲットを当て，速攻ハメ殺す。すかさずバズーカ男の体力を見切ってハメ殺せ。破壊後は，すぐにMAD-BETTYを準備して，出っ張りに隠れたMAD-BETTYを回収しつつ，斬りまくって進んでいくべし。次に，バズーカ男2人を軽くあしらい，奥のメカ魚もMAD-BETTYの残っているうちに破壊するようにすること。すると，鈍臭い奴×3が降ってくる。おそらく1機目まではMAD-BETTYが残っているはず。1機目を破壊したあとにMAD-BETTYが切れたら，すかさずベティーを用意。残りをベティーを飛ばさず，押し当てる感じで破壊しよう。3機目を破壊するとバズーカ男の登場だ。ここはいったん右へ引き返し，ベティーのエネルギーをMAXまで溜めてから戦い始めるように。軽く当ててやってからは，ハメ斬りしつつベティーのエネルギーを溜める。頃合を見計らってベティーを投げ当てて終わりだ。同時に出現するメカ魚は泳がせておけばいい。

ただし，いい加減にプレイできるといっても，いままで解説してきたテクニックを完全に使いこなせることが前提となっている。特にバズーカ男に対して，ハメ斬りのタイミングをミスしてしまうと思わぬ苦戦を強いられることになるだろう。

出現する敵の種類，数は少ないながらも，そこそこ難易度の高いステージかもしれない。



▲斬って斬って斬りまくれ！



▲MAXベティーを押し当てて破壊せよ

3-3 目標：1分10秒

まずは，速攻でベティーを使う。空中に浮かぶ砲台は無視して走り去り，さっさと目の前にある穴に落ちる。穴に落ちたあとは，右にあるゲートを開け，ゲートが開ききる前にさっさと飛び越えよう。

そして，ゲートを飛び越えたらすぐにベティーのエネルギーを溜め始めること。目の前には変な装置があるが，ここのコア(赤い玉と青い玉)がこのステージのターゲットだ。まず赤い玉が現れるので，その場MAXベティー投げで赤い玉を破壊する。次に青い玉がすぐに現れるが，ベティーのエネルギーがMAXに溜まる前なので，1回見過ごしてエネルギーを溜め続け，次に現れたときにMAXベティーで破壊しろ。ちなみに変な装置に密着しようとすると，エネルギー弾をばらまくブービートラップが発動する(上下にある黄色のスイッチ)。勢い余って突っ込むなんていう，限りなくマヌケなことはくれぐれもしないように。

コアを破壊したあとは，ゲートに詰まらないように場所を覚えて，なるべくムダのないようにボスへ向かうだけだ。





▲注意すべき点は特にない。速攻でベティーを使い始め，ひたすら走り，ゲートを開けたあとにコアを破壊するだけだ。かなりイージーなステージなので大ボケをかまさない限り，なんなくこなせるだろう





▶コアを破壊したあとは、ひたすら走り抜けろ！上から降ってくる鉄屑に当たって、タイムをロスするぞ

STAGE3 BOSS

まずエネルギーアイテムを取り，赤，青が分離するまで左端でネバる。そして，敵が攻撃に移り始めたら特攻だ。間合をとりつつ青を丁寧に1斬りずつ，じわじわと押していく。うまくいけば，赤からの稲妻攻撃，地上での攻撃をかわせる。そして，青の体力をある程度削ったら，すかさずEXPLODEで青にトドメを刺す。青の体力を見切らないと反撃をくらうぞ。なお，赤が攻撃する瞬間に無敵時間があるので，タイミングを見切って赤にもダメージを与えておくこと。そしてベティーを用意。次に赤が空を飛んでいるときに，レバー上でMAXベティーを当ててさっさと赤の飾りを破壊する。破壊後，赤は止まっているときに防御をすることがあり，ベティーに対しても無敵になる。これを避け，MAXベティーを中段水平投げで当てて体力を奪え。もしベティーがあるうちに破壊できなければ，間合をとりつつ左壁まで1斬りずつ押していこう。そのあとは注意してハメ斬りを行えばいい。ただし，ベティーがなくなったら相当のロスになることを覚悟すべし。いざというときは左に出現するHI-POWERに頼ることになるが，うまく取るのはなかなか難しいだろう。



▲まずは青を斬る



▲そしてEXPLODEで青を破壊



▲MAXベティーを押し当て……



▲中段ベティー飛ばしでKOだ！

STAGE 4

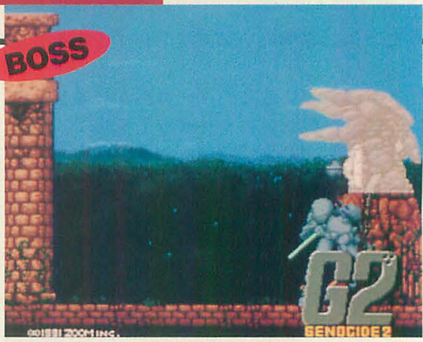
4-1 目標：55秒

ここは，タイミングが合うことが滅多にない苦しいステージ。このステージだけでなく，STAGE4は全体的にミスが多くなりがちで，タイムに影響が出やすい構成になっている。フンドシでもタスキでもいいが，気を引き締めて戦いに挑もう。

まず，ステージ開始直後はベティーを用意し，エネルギーを溜め始める。飛んでるレーザー男①をMAXベティーで破壊し，トンボ②も軽く破壊する。

そして，残りのレーザー男③，④，⑤を倒したら，すかさずHI-POWERを用意し，HI-POWERアイテムを取ると同時に⑥を叩け。⑥はタイミングよく斬れば3段が入るので，3段＋1斬りで殺す。⑦はジャンプ斬りを当てて，一度ひるんだところにもう一度ジャンプ斬りだ。逃すとうっとおしいので確実に破壊せよ。⑨，⑩は，進みすぎると同時に出てくるので，ここは慎重に進みつつ1機ずつ倒していこう。⑪の直前でHI-POWERが切れると思うので，最後は3段×2＋1斬りでトドメを刺す。

まあ，ここは運もあるし難しいステージといえるだろう。



▲石像はベティーの回復に合わせてセンサーを破壊。コアが現れたらMAXベティーを接触させ，そのまま押しつける。ベストタイミングだと，センサー破壊直後にコアが開き，火噴きを省略できる



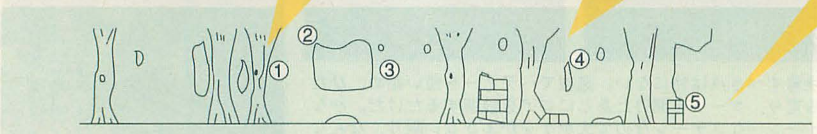
◀MAXベティーで①を破壊。ちょっとベティーのエネルギーを溜めて②を撃破



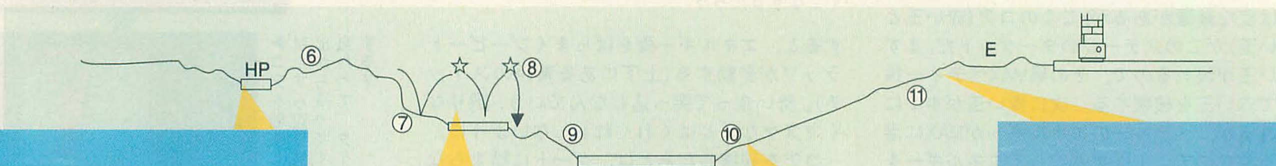
◀④が撃つレーザーに注意。ベティーに当たると溜めがなくなるぞ



◀⑤はいきなり地上でレーザーを撃つことあり。破壊後は，即座にHI-POWERを用意せよ



◀とにかく前半にいるレーザー男が発射するレーザーに注意せよ。もしも，レーザーによってベティーのエネルギーの溜めを消されたら，己の不運を呪うしかない





▲⑥を高速3段斬り＋1発で破壊



◀⑧は連続ジャンプ斬りで料理せよ



◀⑨、⑩は1機ずつ相手をすること



▲右から高速3段斬り×2＋1発だ

4-2 目標：1分8秒

このステージは，初心者イジメ的なかなりいやがらせなステージだ。いったんハマるともう終わり。ちょっと納得いかないところがあるね。

まず，ステージ開始直後に出現するバカ車輪は，高速3段斬りで瞬殺。ついでに目の前にあるエネルギーアイテムも補充しておく。すかさずダッシュすると坂の上あたりでもバカ車輪が出現する。次は前後からくるので，前からくるヤツを飛び越えて，坂下でまとめてケリ殺せ。破壊する前にふっ飛んでいったヤツを追いかけて破壊し，速攻ダッシュで先に進もう。そして，いよいよハマり場が始まる。ここから先は修羅の道，とまではいわないが，いったん失敗すると最初からやり直さなければならなくなる。一瞬のミスが命取りとなることを覚えておこう。進むパターンは右図のとおり。的確な状況判断と正確な操作が要求される。なお，⑦，⑨，⑫′，⑭，⑮は破壊する必要なし。



▲うっとおしいバカ車輪はさっさと片づけろ



▲Fの上にあるEXPLODEは絶対に必要。ベティーを飛ばしてアイテムをゲット！



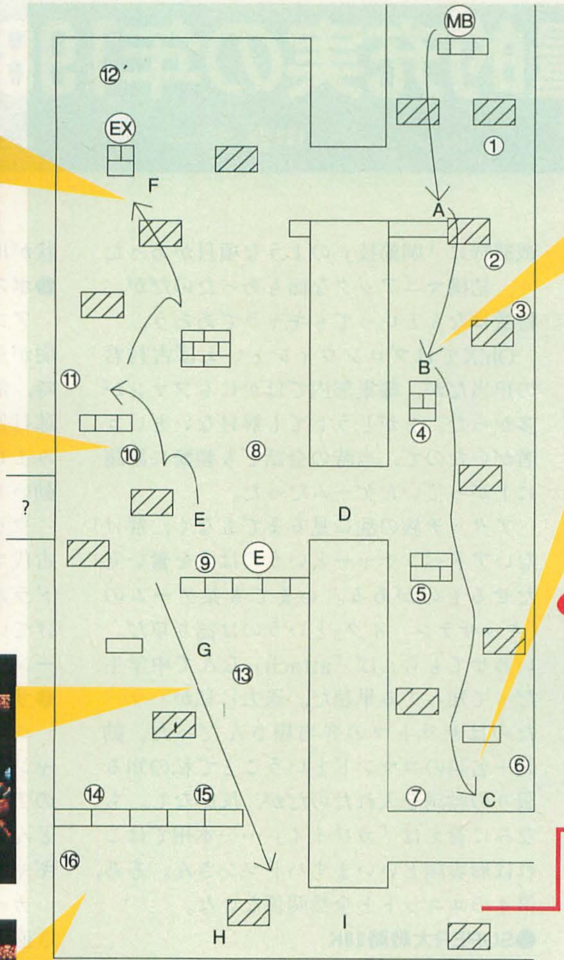
▶E点でベティーのエネルギーを溜め始め、⑪を速攻で破壊



▶⑬の出現位置を覚え、落下中にベティーで破壊できるようにする



▶⑯もMAXベティーを使って、敵がプラズマを吐く前に速攻撃破だ



▲崩れる床の中央に乗り，落ちる途中でMAD-BETTYを準備。落ちるとMAD-BETTYがあるので，使うと同時にゲット。⑥，⑦以外の敵を残すと，うっとおしい状況に陥るので確実に仕留めること。D点から左に移り，⑧をベティーで仕留め，⑪の重力装置をMAXベティーで叩く。そのあとは上に登りEXPLODEを拾え。すぐに落下して，わずかに溜まったベティーを⑬に当てて破壊する。最下段の⑯もMAXベティーで叩こう



◀②、③を破壊するまでB点でネバる。MAD-BETTYを当てるタイミングを覚えろ



◀⑥の重力装置に浮かされたらすぐ上の足場に潜り込み、上昇しないように



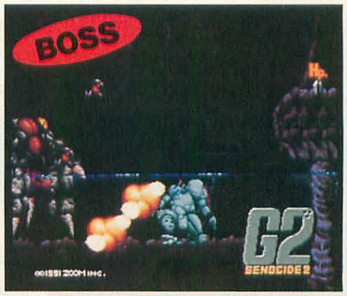
▲ボスはベティーが残っているとラッキーだ。ベティーがないときには，まず速攻でしゃがんで連射をし，ボスを右端に追い詰める。このときボスがヨーヨーを投げてくると完全にヤバイ状況になる。そうなってしまったら根性で右端に追い詰めるしかない。もしも最初で成功したならば，右端で高速3段斬りを使ってハメ殺しを行うだけだ。ベティーが残っていれば，ボスが登場した直後にしゃがんでベティーを重ねるだけで，ボスの第1撃をキャンセルさせられる。そのあとは同じ。まあがんばろう

4-3 目標：50秒

ここに登場するのは，これまた初心者キラーのカニ。G1のときに登場したカニよりはましかもしれないが，かなりやっかいなヤツだ。まずは高速移動落下を駆使してボスの上まで速攻でたどり着くようにする。そして，エネルギーアイテムを取り，待つ。このとき，必ずカニの中心より右にいよう。右すぎてもダメ。落ちていく途中に早すぎず，遅すぎずのタイミングで右端に逃げる。右端でHI-POWERを用意して写真を目安にしてしゃがんで待とう。この位置合わせの意味は，カニが吐き出すガキをここで全滅させることにある。この位置でしゃがみ，連射を使って斬ることにより，バラバラに飛んでくる遠近両方のガキを殺せるのである。

ガキを殲滅したらカニに近寄り，当たり判定が発生すると同時に高速3段斬りでひたすら斬りつける。ここでは，焦らずにしっかりとダメージを与えなければならない。運よくカニの吐き出す弾に当たらなければ，ここでほとんどの体力を奪えるはずだ。次にボスは変形するのだが，変形後も1回分切りつけることができるのを覚えておこう。変形後，左右に飛び跳ねるが，間合をうまく取れば着地ごとに1斬りずつ攻撃を浴びせることが可能となる。ベストパターンは1回目のジャンプで最も右に着地した瞬間に殺すパターンだ。そうでなくても次の変形前には，ボスを殺せるようにしなくてはならない。タイミングは結構シビアかもしれない。

そうそう，殺すときは確実に右端になるように注意しておこう。



▶ガキを倒すベストポジションはこの位置。正確な位置合わせが要求される場面だ



▲弾に当たらないことを祈ろう



▲殺すときは必ず右端で

質的量的充実の年1989年

Nakano Shuichi 中野 修一

ごめんなさい

最初にお詫びを。10月号で紹介したデスブリンガーは1989年の作品だということが判明した。かなり初期の作品だと記憶していたので間違ってしまったようだ。だって，確認でマニュアルを見たら1988ってコピーライトが入ってたんだもん……。

また，沙羅曼蛇で1周に出てくるフォースフィールドは3個であった。これは単純ミス。ちなみに，沙羅曼蛇にはそうとう入れ込んでいた八重垣氏はアーケード版で一度だけ2個のフォースフィールドをつけたことがあるそうだ。プログラム的には2個以上つけても最初の奴しか防御の役には立たず，かといって当たり判定がないわけでもなく容赦なく削れるだけ削れていく……という涙の出る仕様らしい。オートレベルコントロールだから，フォースフィールドをつけると敵の攻撃はいっそう激しくなるのはいうまでもない。

それはさておき，1989年に発売されたゲームはとても多い。今回は私の手だけでは間にあいそうにないので，アシスタントとして須藤芳政君に協力してもらった。それでもページ数と時間の制限でこぼれた作品も非常に多いことをお断りしておく。

記憶があやふやなので，順番などはかなりいい加減になってきている。内容にかなり偏りがあったり，極度に私的な事柄についての話になったりすることもあるが，まあそれはそれということでご容赦願いたい。

1989のゲームたち

●第4のユニットシリーズ

コマンド選択式のアドベンチャーゲーム。戦闘時に「関節技」のような項目があったり，結構マニアックな面もあったのだが，特徴はなんといってもキャラであろう。

Oh!Xではブロンウィンといえば古村君の担当だが，編集部内でほかにもファンが多かった。2がどうしても解けないという者がいたので，当時の会話でも頻繁に話題に上がっていたゲームだった。

アタッチ族の乱に見るまでもなく，解けないアドベンチャーというのは人を奮い立たせるものがある。いまでも某ゲームの「キッサテン　イク」というのは語り草だ。いわせてもらえば「attach」なんて中学生だって知ってる単語だ。過去に私がハマったのはサラトマの弁当屋さんだった。動詞＋名詞のコマンドということで私の知る限りの動詞を入れたのだが，反応なし。ちなみに答えは「カワイイ」……本州ではこれは形容詞といいますハドソンさん。ああ，第4のユニットと全然関係ないな。

●SUPER大戦略68K

ウォーシミュレーションの代名詞といえる大戦略シリーズのX68000版。画面やサウンドなどを見ると結構ノリがいい。処理も速く，テンポよく操作できる。当時最新であった大戦略IIではなかったのがちょっと惜しいところだ。

●スタークルーザー

3D画面は256×256ドットだったが，エッジをアンチエイリアスしているのが良心的。

ポリゴン技術，音楽，独特のテイストのあるシナリオにシビレたファンが多い。

PC-9801版のスタークルーザーIIを見ていると，X68000版のスタークルーザーIのほうがデキがいいという気もする。

どう考えても完全互換のはずの倍速アクセラレータH.A.R.P.で，オープニングの2ループ目に限り計算を間違えるという症状が出ていた謎のソフトでもある。

●ボスコニアン

アフターバーナーを移植していた松島少尉が息抜きに作ったものと伝えられる。当時，電波のスタッフがアフターバーナーの進行具合を見にいった際，「こんなの作ってみました」といわれて画面を見るとあれが動いていたという伝説がある。

ウリはなんといっても音楽である。永田，古代コンビの名曲と初めてAD PCM同期ドラム採用のドライバが比類ない成果を上げている。その後のX68000ミュージックシーンに大きな影響を与えた作品である。

●ザ・キングオブシカゴ

シネマウェアといういまは聞かれないジャンルのゲーム。要はマルチエンディングのアドベンチャーゲームだ。選択式でどんどんストーリーが変わっていく。チンピラギャングからシカゴの帝王への道を目指す。

カートゥーン調の原作をX68000用に描き直している。なかなか味はあるが，ものすごく遅かった。

●プロダクションマネージャー

芸能プロダクションの社長となってアイドルを発掘するという「育てゲー」である。オーディションで人を選び，芸名（ジャンルによって向き不向きがある）をつけ，日々お稽古その他で各種技能を磨く。

ざっと見たパラメータも多いのだが，さらに内部パラメータというものがあり，この「内部パラメータが見えない」というのが面白みでもあり，ガンでもあった。

幸い，編集室のサンプル版には内部パラメータのアンチョコがあったので私はそれなりに楽しめた。というか，パラメータはかなり複雑で，アンチョコを見ながらでも全貌はなかなかわからないという具合だったのだ。キャンペーン，アンケート，根回



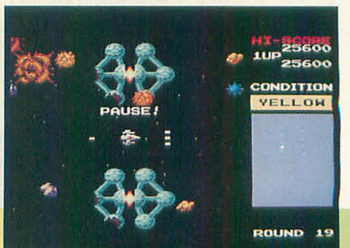
第4のユニット。1～3が発売された



兵器の設定も細かい



スペオペ仕立てのスタークルーザー



豪華になったボスコニアン

[1989]


ザ・キングオブシカゴ。キャラは動く


これがプロダクションマネージャー


カサブランカに愛を。渋い


R-TYPE。グラフィックは綺麗

し，賄賂，内部データには生理日まで設定してある……普通にやるには複雑すぎたな。

コツがわかればそこそこ楽しい。毎日毎日時代劇映画の主演の仕事（いちばん儲かる）が入ってくる女優を育て上げた。が，パラメータが最大値（と思われる）になった途端，まったく仕事がこなくなった。オーバーフローか，人間国宝の域に到達してしまったか……。

もうひとつの遊び方がある。キャラの状態によっていろいろな仕事が入ってくるのだが，よい仕事ばかりとは限らない。気にいらない仕事をやらせようとすると，すぐに事務所をやめてしまったり，各種パラメータに悪影響が出てくる。普通は選ぶべきではないのだが（ちょっとやそっとじゃ引き受けてくれないし），選択肢にはそういった項目が存在する。

元がアイドル志望のタレントだけあって，
「私もうこんな仕事できません！」
とかワガママなことをいってるのを脅したりなだめたりして対応していく。とりあえずヌードダンスっと。
「私，この仕事したくないな……」
とりあえず，なだめすかせてみる。
「もうがんばっていく気になれません」
リセット。と，1歩ずつ着実に女衒屋稼業の道を進んでいくこともできる。

あまりにも無名のソフトだったが，もう少し作り込めば非常に面白いものになったのではないかと思わせる逸品であった。5年早すぎたか。

●カサブランカに愛を

シンキングラビットのコマンド直入力式アドベンチャー。モノトーンのグラフィックが渋い。オープニングでは多重スクロールなんかも見せてくれるが，ほかは目新しいと思われるものはなかった。（Y.S.）

●R-TYPE

処理が遅かったり，爆発パターンが違ったり，ミサイルの追尾が弱かったり，妙に難易度が上がっていたりと，難点はいろいろあり移植もののデキとしては最低レベルにあるものの，原作と比べなければそれなりに遊べる。やっぱり元が名作だったんだなあと思わせる。PCエンジン版に比べてもRGBで見るグラフィックは美しい。

●サンダーブレード

シャープ/SPSによる大型作品。2D，3Dを織り交ぜたシューティングゲームだ。

移植の難しさではアフターバーナーを上回ると思うのだが，どうにか動くものに仕上げている。一部の簡略化は残念だが，しかたないところであろう。原作では川を登っていく面が好きだったのだが……。

ちなみにポーズをしてもちゃんと止まってくれないので画面撮影は至難の業だった。

●アフターバーナー

当時の業界で最高のビッグタイトルのひとつ。X68000に移植されることが話題となっていた頃，当時発表された富士通の新機種FM TOWNSが対応ゲーム第1弾として真っ向からぶつけてきた（以下日記風）。

当時，Oh!FMのスタッフを見ていると，FM TOWNSの発表会から戻ってくるや，オープニングで映写されたTOWNS版アフターバーナーの映像を評して，
「完・璧！」
と珍しく興奮していた。偵察がてら，秋葉原で流れ始めたTOWNS版の店頭デモを見て回る。と，偶然，当時受験を終えたばかりの石上君に出会った。
石「いやー，凄いことができるようになりましたねぇ」
中「え，そう？」

1年以上禁コンしていた彼にはカルチャーショックだったようだが，予想よりもカクカクしているのを見た私はスキップしながら帰路についていた。

で，X68000版だ。決して完全移植というわけではないが，ゲームの魂をよくつかんだ移植として名高い。当初「マウス対応」というのを聞いて不安にかられたものだったが，フタを開けてみるとマウス対応は大正解だったといえるだろう。シャープからはこれにあわせたかのようにアナログジョイスティック，サイバースティックが発売された。音楽は多少もの足りなかったのも確かだが，その後，MIDI対応パッチなどが発表されより完成度を上げていく。

その後長期にわたりアンケートハガキの「最近買って気にいったソフト」欄を独占し，集計の浦川君を辟易させていた。とにかく，圧倒的な支持を得た作品であった。

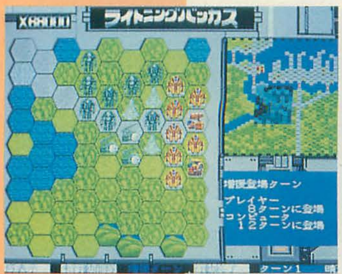
●アークス

ウルフチーム参入第1弾だが，最初はちょっとひどかった。X68000に慣れていないのでベタ移植はともかく，色抜きに失敗しておりオープニングで口の中に空が見える


サンダーブレードの3D面


スピード感が命のアフターバーナー


パックマニア。パックマンのリメイク



SF仕立てのライトニングバッカス



ソフトでハードな物語2。ここは……



キャラは可愛いんだけどねえ……



ボスキャラ攻略が楽しい

などの粗さが目立った。やがてくるテクニカルな作品群の片鱗はまだ見えない。

●大海令

アートディンクの海戦シミュレーションゲーム。プレイヤーは基本的に艦隊の司令官を担当し，現場の処理はすべてコンピュータが行う。かなり自由度が高く先進的なシステムではあったのだが，あまり内容を知られることがなかったのが惜しい。ウォーシミュレーションというのは作り込まれれば作り込まれるほどマニア以外には敬遠される傾向にあるようだ。

●パワーリーグ

元はセガのアーケードゲーム。それをハドソンが移植している。内容は割と普通の野球ゲーム。高橋名人が声を担当していた。

●パックマニア

パックマンを疑似3D化しクォータービュー視点で表示したもの。元祖パックマンのようなものがオマケでついていた。一部ロートルからは完全移植のパックマンがほしかったという声が上がっていた。

●フルーツフィールド

その昔Pioという雑誌があってその誌上で一挙5機種ぐらいに対応したプログラムリストが公開されたものがまさかX68000へ移植されるとは思わなかった。

矢印形のブロックを移動させてフルーツをむさぼるという「倉庫番」の変形版とでもいうべきものだが，「いまさら……」という気もしないではなかった。　(Y.S.)

●雀豪1

麻雀ゲームとしては評価の高い雀豪シリーズの移植だ。ただ，X68000に慣れていないのか，かなり激しいバグが残っていた。編集部では手牌のなかに捨て牌用の小さな牌が交じっていたというβ牌事件で有名である。原作がよかっただけに残念だった。

●ライトニングバッカス

私はこのゲームはやらなかったのだが，手を出した人が残らず絶賛していたのが印象に残っている。X68000らしいシミュレーションゲームとして評価の高かった作品だ。ノリがいいらしい。

●ソフトでハードな物語2

業界ネタアドベンチャーの第2弾である。

かつて古村君がこれほどレビューで苦しんでいる姿は見たことがなかった。どうやってもクリアできないというのだ。

問い合わせたところ，クリア条件が非常にシビアで，なおかつ，20回に1回くらいしかクリアできないようになっているとか。クリアした人はいるのだろうか？

●ニュージーランドストーリー

このゲームがゲームセンターで現役だった頃，主人公の頭に斧がもろ突き刺さって苦しみもがいたり，黒焦げの焼死体になってしまうなどの残忍な場面をデモ画面でじっと眺めている人は多数見受けられた。しかし，難易度が高いせいか，大金を注ぎ込んでクリアできるようになっても自慢のネタにならないせいなのか知らないが実際にプレイしている人はほとんどいなかった(私の実家のほうだけかもしれない)。私も一度やってみたが，即ゲームオーバーになってしまい以後二度とやらなかった。

そのニュージーランドストーリーのX68000版を起動した私の目へ最初に飛び込んできたのはセーラー服。なぜ，いきなりセーラー服の女子学生なんだろうか？　と当時は議論を呼んだ。

で，実際のゲーム内容。なんとも気分的にスッキリしない主人公の死に様にコンティニューせずにはいられなくなり，意識もうろう状態で徹夜した末になんとか最終面へたどり着いたのだが，ツルツル滑る氷のステージ設定に胃液の大量分泌を感じ自らドクターストップをかけた問題作である。誰もがやめるにやめられず，半ば中毒にかかってしまうゲームだった。　(Y.S.)

●ファンタジーゾーン

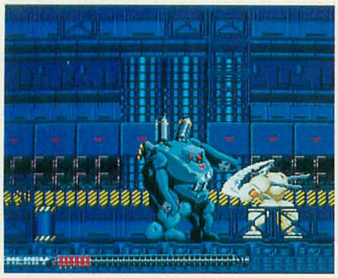
セガの名作シューティングゲームの移植版。パステルな色調に個性的なボスキャラ，使いどころを考えられたウェポンシステム，難しすぎない難易度とバランスのよい仕上がりのゲームだった。女の子とかもよくやってたな。個人的には「お金を拾う」というシステムに燃えていた気もする。

忠実＋αの要素をてんこ盛りし，移植はこうありたいと思わせる模範作品だ。デモプレイがなかなか凄いし，リプレイデータをファイルに残せたり，BGMは国内版，海外版それぞれにオリジナルバージョンとアレンジバージョンを用意，3D眼鏡対応といったこれでもかという豪華装備。さらに隠しオリジナル面としてスペハリ面なども用意されていた。移植ものの金字塔である。

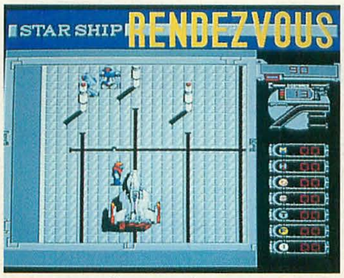
●ジェノサイド

ズーム衝撃のX68000デビュー作，めちゃくちゃ固い敵を賢いロボット「ランディ」に乗った主人公がザクザク斬りまくるアクションゲーム。ランディよ，そんなに賢いなら攻撃も全部ランディがやってくれたまえ。気合の入った演出や巨大なキャラクタの動きは，アーケードからの移植ものばかりが目立っていた当時のX68000アクションゲーム中で異色な存在だった。しばらく遊んで気持ちが加熱してくるとステージ間の登場人物の会話のやりとりが邪魔臭くなってサルのようなジョイパッド連打でキャンセルしたくなってしまうのがちょっと不満。音楽はのちに発売される2よりもこちらのほうがいいって人もいるのでは？　……いませんか，ハイ。　(Y.S.)

[1989]



ジェノサイド。敵はデカくて固い



スターシップランデブーのゲーム画面



ネジをつけただけでは終わらない



デカキャラ高速表示のメタルサイト

●スターシップランデブー

アイテムを集めて実写のマカダム，というエロゲー。評価は「やりたいことはわかるが，モデルの人選を誤っている」のひと言に尽きる。

●ねじ式

つげ義春の世界を題材にしたツァイト制作のアドベンチャーゲーム。ゲームのシステムなどは露骨にフラグ探しであまりほめられたものではないが，世界観の広がりや完成度でこれに勝る作品はまだ見たことがない。全然関係ないが，Dの食卓ってなんであんなに態度がデカいんだろう。ありきたりなゲームだと思うのだが。

●メタルサイト

システムサコムの3Dシューティングゲーム。FD5枚組で入れ替えが面倒だなんて当時のX68000ユーザーはいまほど贅沢ではなかったのでボヤいたりはしなかった。

画面の状況を把握するのには多少の鍛練が必要である。鍛練を積んだとしても敵が画面に溢れるといまなにが起こっているのかわからないのでとりあえずグルグル自機を回転させてみるという,「攻撃をかわす」というよりも「回転作業する」という感覚が強かった。3人のキャラクター選択のメリットはほとんどなかった。　(Y.S.)

●ガウディ・バルセロナの風

ウルフチームのアドベンチャーゲーム。同社のアドベンチャーはこれっきりで，あとはRPGやアクションへ流れてしまうため貴重な逸品。付属のマウスパッドを手に入れるためだけでも買う価値はあった。パッケージにある「ヒステリー・リピート・システム」はたぶん誤字である。 (Y.S.)

(編注：これ以外にも「あ～くしゅ」というアドベンチャーもありました)

●ミッドガルツゴールド68K

このあたりになるとウルフチームのオープニングデモは本領を発揮してきている。外人声優を使ったアニメ調のデモは浸りきっていてなかなか恥ずかしいものがある。やりたいことはよくわかるんだけどね。こういうのはPC-8801やPC-9801でやりたくてもできなかったんだろうなあと思わせる。

セイクリッドファンタジーシリーズ(SFS)の一翼を担い,「ゴールドはシューティング」ということでゲーム内容はシューティングゲームとなっている。

パッケージは非常に豪華でセンスもいい。マニュアルも豪華だが，安彦良和調のあまりうまくないイラストや60ページに及ぶ設定資料……。やりたいことはよくわかるんだけどね。

●アルフェイム

ご存じザインソフトの代表作だ。すでに説明の必要はないと思うが，当時のOh!Xを見ると新作コーナーの紹介者も困ったらしく「ザインソフトからはディオスとアルフェイムが登場です。2つともザインらしさにあふれていますね」で片付けている。

●シャッフルパックカフェ

結構熱くなれるエアホッケーゲーム。このゲームでマウスを壊したのが若干1名。女王様(?)の攻撃を見切れるかどうかがポイントか。人対人の対戦がほしかった。

●リングマスター

本格指向RPG。だが，NPCの会話を人工無脳にしたのはやりすぎかもしれない。ハードディスクインストールOKでマニュアルプロテクトになっていた。が，たいていの人は臙脂色の紙に藍色のインクで書かれた魔法書を見てうんざりする。

●ガンマプラネット

ワイヤーフレームのタンクバトル。多段ジャンプ抜きのワイヤーのジオグラフシール，というとかえってわかりにくいか?

ワイヤーってのも味があっていいものだが「音楽の評価が高い」ということでかろうじて知られている作品だ。

●夢幻戦士ヴァリスII

ゲーム内容はともかく，着せ換えモードやかなり恥ずかしいビジュアルシーンなど，好きな人は好きなゲーム。音楽はイカしているぞ。内容の軟派さにしてはゲーム部分がきついのが難点か。

このゲームが出てくると編集部では必ず



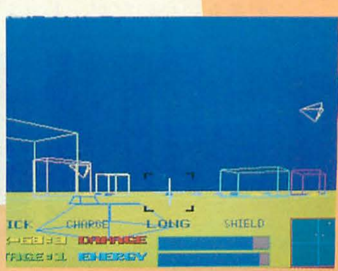
やられ声が悩ましいミッドガルツ



得体の知れない対戦相手ばかりだ



本格RPGのリングマスター

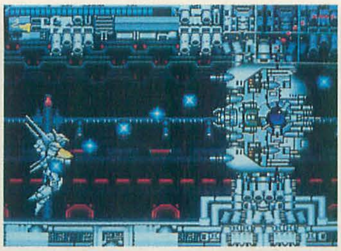


ワイヤーシューティングが新鮮

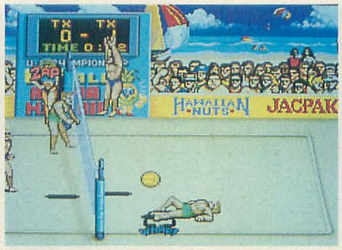
[1989]



ヴァリスII。ゲームは結構シビアだ



ユニークな疑似3Dシステムではあった



ビーチバレーってのがイカすV'BALL



モトス。相手をはじき落とせ！

「あのときデビューした麻生優子はどうなったのか？」という話題になる。

●A-JAX

コナミブランドで出したX68000対応ソフト第1弾。アーケードからの移植で2D, 3Dと2種類のシューティングゲームを楽しめる。発売当時X68000ユーザーは目を輝かせながら買いに走った(と思う)。

デモ画面で見る3Dシューティングモードは「凄い！」のひと言で，実際やってみると別の意味で「すっご～い！」と感じ，このゲームのメインは2Dシューティングモードなんだなと誰もが悟ってしまう。シューティングゲームらしからぬ妙に長いディスクアクセスが気になった。　(Y.S.)

●ナイトアームズ

アルシスが疑似3Dスプライトシステムを引っ提げて送り出したアクションシューティング。画面の奥行きを生かした攻撃などもあったのだが，基本的には横スクロールシューティングゲームのようなものだ。ボスキャラとかを見ると……いや，趣味に走っているなあ。

3Dスプライトといっても，説得力があるのはオープニングの最初だけでゲーム中はあまり関係ない。ちょっともったいない。

●V'BALL

ドッジボールに続く，テクノスのスポーツゲームはなんとビーチバレーだった。操作もわかりやすく，対戦が結構熱いし，そこそこデキもいいのだが，ドッジボールほど燃えなかったのはなぜだろう。

●モトス

渋めのタイトルである。おはじき落としのようなルールのアクションゲーム。OSから作り直して超高速ブート……というのがウリのひとつだった。目標値は達成できなかったようだが。

MIDI音源でコルグM1にまで対応していたのが印象に残っている。まさかプログラムとコンビネーションデータが消されるとは思わなかったしなあ。

●フラッピー2

かつての名作が疑似3Dになって帰ってきた。個人的に前作はパズルゲームの最高峰だと思っているが，2はちょっとイマイチ。X68000らしさを出そうと色気を出してクォータービューにしたのだろうが，明らかに失敗している。パズルゲームで全体を見渡せないというのは非常につらい。

●スーパーハングオン

デキはいい。というか，ちゃんと遊べる数少ないレースゲームだ。ラスタースクロールによるアップダウンの実現が云々と，いま考えると平和な時代だったなあ。

●殺人倶楽部DX

リバーヒルソフトの代表作をX68000用にリメイクしたオリジナルデラックス版。英語，日本語対応の2モードバイリンガルシステムがウリであった。「ワオ，コ～レハ～，エイゴノ，ベンキョニ，ナリマ～ス」といいながら辞書を片手に最後までやりとげた人はたぶんいないと思う。

オープニングはカッコイイ。しかし，「DX」のネーミングは当時「がんばって～がんばって～仕～事！」のCMでお馴染みだった某栄養ドリンクの名前からパクってきたように思えてならない。　(Y.S.)

まとめ

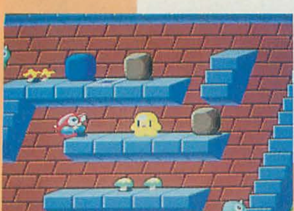
とにかく玉石混淆，もの凄い技術と情熱で仕上げられた作品があるかと思うと人をなめたようなものも垣間見られる。しかし，PC-9801からのベタ移植のようなものはほとんどない。X68000用オリジナル作品がかなり多いという点が重要だ。この時期の作品は，作る側もある種の思い入れを持っていることがよくわかる。

作りたい作品を作るのはよいことだが，それがユーザーにとってもありがたいものだったのかどうかは，また別の話になる。

凄いソフトが出てくると，ユーザーには「これくらいのことはできるマシンだったんだ」という認識が生まれてくる。ゲームに対する要求もエスカレートしていく。

売れるタイトルはかなり売れていたのも事実だ。市場シェアでいえばPC-9801の1/30～1/50程度しかない機種で売り上げの上位に入るというのは，常識では考えられない。この時期「ちゃんと作られたソフトはちゃんと売れる」という図式が確立されていたのは非常によいことであった。のちに，この「ちゃんと作れば……」というのを誤解してくるソフトハウスも多かったそうだ。「ちゃんと作れば1万本近く売れます」といわれると，なぜか皆さんうれしそうな顔をされたそうで……。

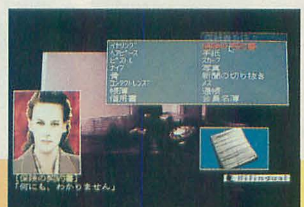
しかし，実質の市場規模は決して大きなものではない。これ以降の勢いは非常にバブリーなものだったともいえよう。



全画面見えないのが悲しい



この程度のカーブでもハングオン

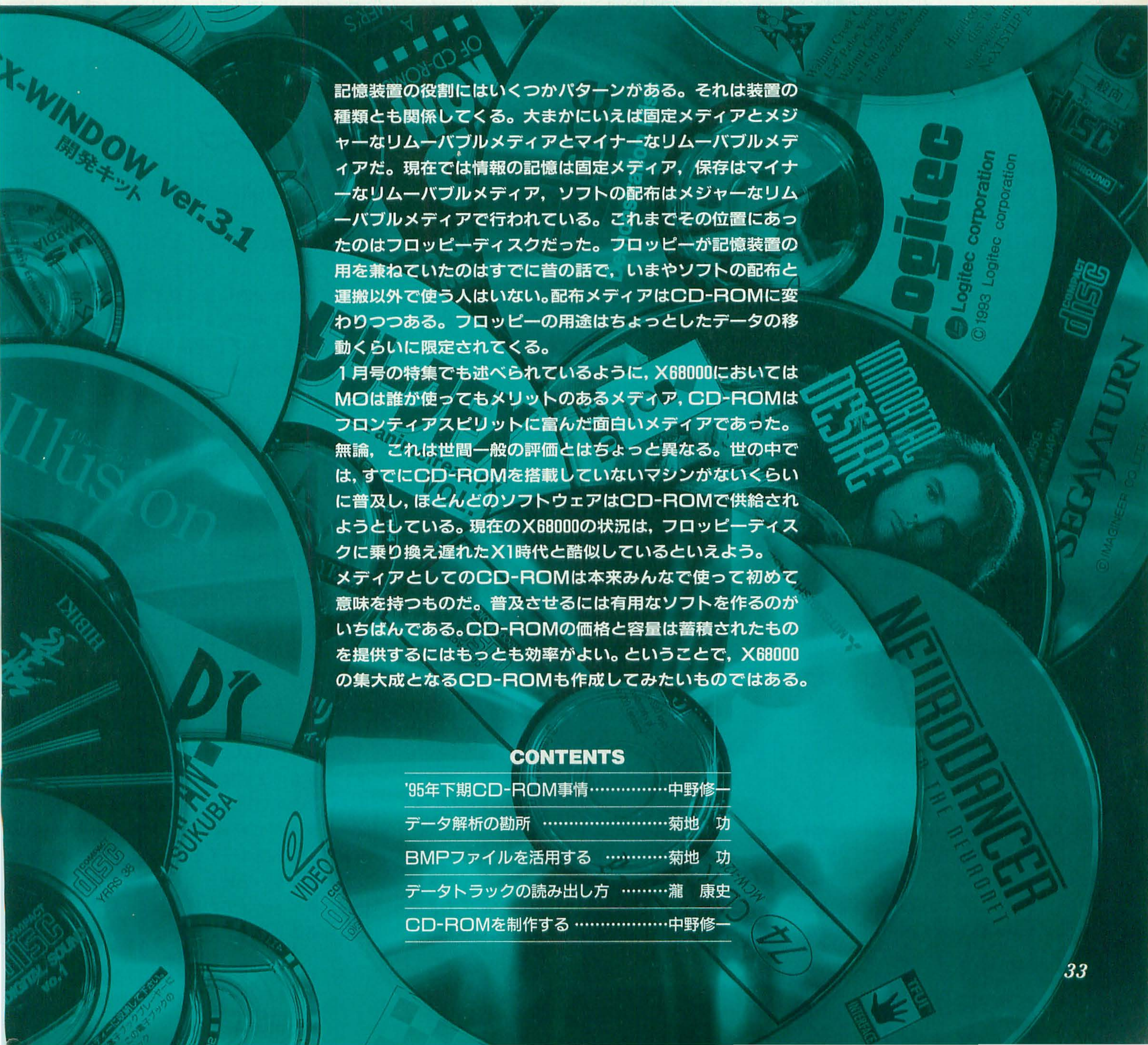


写真取り込みがいい感じだ



[特集]

はりきって使う CD-ROM



記憶装置の役割にはいくつかパターンがある。それは装置の種類とも関係してくる。大まかにいえば固定メディアとメジャーなリムーバブルメディアとマイナーなリムーバブルメディアだ。現在では情報の記憶は固定メディア，保存はマイナーなリムーバブルメディア，ソフトの配布はメジャーなリムーバブルメディアで行われている。これまでその位置にあったのはフロッピーディスクだった。フロッピーが記憶装置の用を兼ねていたのはすでに昔の話で，いまやソフトの配布と運搬以外で使う人はいない。配布メディアはCD-ROMに変わりつつある。フロッピーの用途はちょっとしたデータの移動くらいに限定されてくる。

1月号の特集でも述べられているように，X68000においてはMOは誰が使ってもメリットのあるメディア，CD-ROMはフロンティアスピリットに富んだ面白いメディアであった。無論，これは世間一般の評価とはちょっと異なる。世の中では，すでにCD-ROMを搭載していないマシンがないくらいに普及し，ほとんどのソフトウェアはCD-ROMで供給されようとしている。現在のX68000の状況は，フロッピーディスクに乗り換え遅れたX1時代と酷似しているといえよう。

メディアとしてのCD-ROMは本来みんなで使って初めて意味を持つものだ。普及させるには有用なソフトを作るのがいちばんである。CD-ROMの価格と容量は蓄積されたものを提供するにはもっとも効率がよい。ということで，X68000の集大成となるCD-ROMも作成してみたいものではある。

CONTENTS

さらなる活用のために

'95年下期CD-ROM事情

Nakano Shuichi 中野 修一

**長らく縁がなかったもののこのところ急速に普及しつつあるCD-ROMドライブ
有効に活用するにはどういったものを導入すればいいか
ここでは最近のX68000をめぐるCD-ROM事情をまとめてみよう**

今年の初めにOh!Xで初めてのCD-ROM特集をやったのだが，1年弱でかなり状況も変わってきた。前回がCD-ROMユーザーがようやく増え始めた時期とすれば，いまや普及期にある。X68000ユーザーにおける，5人にひとりくらいのMO所有率というのも驚異的だったのだが，アンケートハガキの所有機器状況を見ても最近はCD-ROMがそれを凌駕するようになってきたのだ。

他機種では普及率が100%に近くなっていることを思えば（デスクトップタイプでCD-ROMドライブを標準搭載していないマシンはほとんど存在しない），まだまだな状況だが，対応ソフトがほとんどないことを考えるとこれも驚異的なものといえるだろう。

急激な普及の鍵になったのは，EX-SystemがCD-ROM対応すると発表したことや，CPKPLAY.X，「SX-WINDOW ver.3.1開発キット」といったCD-ROMが活用できる環境が少しずつ揃い始めていることが挙げられるだろう。ソフトバンクの責任重大といったところだ。

さて，いろいろと期待をしてCD-ROMを導入していく人が増えるのは喜ばしいことだが，心配なのは「ちゃんと使っているか？」ということだ。前回のCD-ROM特集でも指摘したとおり，対応ソフトがない場合のCD-ROMドライブは積極的に使っていく気持ちがないとあまり面白い機器ではない。

「あなたがいちばんよく使っているCD-ROMソフトはなんですか？」
「スーパーリアル麻雀PV」
というのではちょっと悲しい。

いまのところCD-ROM使用の目的となるのは，Windowsのグラフィックデータであったり，PhotoCDであったり，サウンドデータの取り込みであったり，場合によってはゲームの隠しメッセージを読むためだったり，動画データの鑑賞であったりするわけだ。買ってきた雑誌にCD-ROMが入っていると覗いてみようかという気になるだろうし，パソコンショップに使えそうな壁紙集や素材集があれば誰だって使ってみたいと思うものだ。それが基本的な使い方である。

次に挙げられるのが，特定のアプリを使って大量のデータを活用していくこと。辞書関係がこれにあたるだろうか。常用できるような辞書やデータがあれば非常に有効な使用法といえるだろう。

次に解析である。他機種用のプログラムが必要なデータなどはX68000では使用できない。しかし，それがどういうふうに作られているのかを調べたり，データ構造を調べてみたりといった個人的な解析というのはパソコンらしい楽しみ方のひとつだ。

作る側からすれば解析されて喜ぶ人はいないだろうが，ゲーム機のデータなどはけっこう面白い素材だ。ものによってはソースコードの一部が入っていたりすることすらある。多少危ない分野ではあるが，解析禁止となっているものを除いて個人で楽しむ分にはそれほど問題はないだろう（PlayStation陣営の「解析禁止」というのは，在野の人材を登用しようとしている割には不粋な方針ではある）。

最後に，究極のCD-ROMの使い方として，「ないのなら作ってしまえ」を実践する道もないではない。CD-ROM対応ソフトがなければ作ればいい。対応CD-ROMがなければ作ればいいというわけだが……。これもいまや決して不可能なことではない。

しかし，CD-ROMを作る側になって詰め込んでみると，600Mバイトというのがいかに広大な空間かというのがわかる。EX-SystemでCD-ROMを付属させるために焼かれたライトワンスCDを見るとデータが書き込まれているのはほんの一部にすぎない。実際のところ，容量だけを見ればMOが最適なのだが，MOをプレスする業者というのは聞いたことがないうえ，メディア単価が違いすぎる。

記憶メディアの乱立が続くなか，大半のものが将来的にはDVDで統一される可能性もあるものの，データ配布用媒体としてCD-ROM以上に効率がよいものは当分出てこないだろう。

まずはドライブ

まず，CD-ROMドライブを導入しなくてはならない。

最初に問題になってくるのは，どこのドライブを使用しているかということだ。CD-ROMでは総合的に見て東芝ドライブがもっともおすすめできる。おすすめできないのがNECドライブで通常のSCSIコマンドでは音楽CDの演奏ができない場合がある。あとのメーカーはそれほど大差はない。音声データの取り出しという特殊な処理まで求めるならこのあたりは注意しておかなければならない。

同じ系列の製品でもマイナーチェンジで型番がちょっと変わったり，甚だしい場合は同じ型番のままロットによって内蔵されているユニットが変わってくることがあるので注意が必要だ。型番に"A"がつくだけでドライブユニットが別メーカーになっていたりするのだ。

しかし，どんなドライブでもSCSI仕様であればデータの読み出しくらいはちゃんとできる。

最近は4倍速が標準で，6倍速から8倍速というものまで現れてきている。遅いより速いほうがいいというのは確かだが，4倍速ドライブで転送レートが秒間600Kバイトだ。SCSI2ボードを使わない限りこれ以上高速なものを使ってもほとんど意味はない。単にメディアが読めればいいというのなら倍速ドライブで十分だ。巨大なグラ

フィックデータをメインに扱うというのなら4倍速のほうがいいだろう。

次にドライバ。これはもう計測技研のCD-ROMドライバに尽きる。またフリーソフトウェアでCDDRV.SYSというものも出回っている。多少機能は縮小されているようだが，普通に使うには問題はないだろう。

さらに「SX-WINDOW ver.3.1開発キット」にはフリーウェアを集めたCD-ROMが付属しているのだが，それを動作させるためのドライバとして計測技研からCDDEV.SYSの機能限定版が提供されている(これは5インチFDに収録)。いまならCD-ROMドライバよりも入手しやすいかもしれない。

機能限定版の制限事項を挙げてみよう。

1)　オーディオCD再生に対応していない

音楽CDをパソコンで鳴らそうとする場合は問題になる。単にCD-ROMとして使うなら別になくても困らない。

2)　キャッシュ指定ができない

CD-ROMは遅いとはいっても，転送速度は問題なく，シークが遅いだけだ。データ部分とは離れたところにある割には頻繁なアクセスが必要なFATやディレクトリ部分をドライバ側が管理すれば，アクセス速度を向上させることができる。そういった機能がver.2.1から装備されていたわけだが，簡易版ではこれが省略されている。

キャッシュバッファの指定がなくなっているので，多少アクセスに時間がかかるかもしれないが，致命的な問題ではない。

3)　多連装ドライブに対応していない

そういうドライブでも最初の1ドライブはちゃんと認識される。普通のドライブであればまったく問題ない。

4)　サポートソフトがついていない

CDプレイヤーや開発用ライブラリは必ずしも必要ではないが，Macintoshファイルアクセスツールがないのはちと痛い。それでもWindows用CD-ROMなどだけを使うにはまったく問題がない。

といった具合に，オマケとしてはかなり太っ腹な仕様で，機能限定版とはいえ非常に使えるものになっている。まだCD-ROMドライバを持っていない人はとりあえずこれを使ってみるのがいいだろう。おそらく，MacintoshのCD-ROMを扱う必要が出てくるまではこれで十分であろう。

X68000で使えるCD-ROM

X68000で使用可能なCD-ROM（メディアのほう）を確認してみよう。

「SX-WINDOW ver.3.1開発キット」が発売され，CD-ROMが付属したことでSX-WINDOW周りのフリーソフトウェアは大量に提供された。開発関係には興味がなくともSX-WINDOWユーザーは必携である。

フリーソフトウェアセレクションvol.1，2は残念ながらすでに売り切れ状態となっている。よって，専用といえるのはソフトバンクから近刊予定のNetBSD本，EX-Systemのムックを含めた書籍類のみということになる。

あとはWindows用のグラフィックデータ集であったり（Macintosh専用は避けたほうが無難），PhotoCDであったり，基本的に機種を問わないデータというものがいくつかある。それらは使い方次第でCD-ROMを有効に活用する材料になる。

そういったもののひとつ，辞書関係を少し詳しく見てみよう。

電子辞書関係

X68000対応ソフトがほとんどないとはいえ，電子ブック用の辞書などはアクセスソフトさえあれば立派に使うことができる。

こういった電子辞書の代表は「CD-ROM広辞苑」だろう。SX広辞苑やコマンドラインからの検索では電脳倶楽部vol.44収録のKOJIEN.Xが使用できる。

広辞苑も悪くないが，やはり手軽に試せるのは電子ブック用の辞書である。アクセスソフトとして，計測技研から発売されているSX広辞苑ver.2とシャーペンワープロパックのおまけでLIGHTWING（以下LW.X）といったものや電脳倶楽部vol.89に掲載されたEB.Xといったものがある。

LW.Xでは，たまにアクセスできない単語があるので注意が必要だ。これらの辞書では検索キーがあらかじめ用意されているのだが，いくつかの辞書で必要以上にアクセスを繰り返し異様に遅くなることもある。

EB.Xで間にあう作業も多いので，まずEB.Xを入手して行うこともおすすめする。SX-WINDOW上では，コンソールをひとつ開いておけばお手軽な検索システムができあがる。

ただし，EB.Xも万能ではない。EB.Xでは，「現代用語の基礎知識」で見出し語しか出てこないという症状が出ており，逆にLW.Xでは「漢字源」で解説文が出てこないという症状があるので使い分けも必要かもしれない。ま，EB.Xはフリーウェアだから，こう書いておけばそのうちバージョンアップされるだろう。

こういったCD-ROM辞書はCD-ROMドライブを有効に使用するための強力な武器となるだろう。アクセスソフトさえあれば，市販の辞書類を買ってくるだけでそのまま使える。電子ブックで想定されているいくつかの機能は使用できないが，主要な用途である情報検索にはこと欠かない。また，辞書類以外のものではあまり有効な動作は期待できないが，一応動く，という程度には情報を引き出せる(おすすめはしない)。

たまにLW.XでもEB.Xでもうまく表示できないものもある。検索キーとなる情報が登録されていなかったりするものもあるのだ。そういったときのために電子ブックの内容をダンプするツール（というほどのものでもないか）を作ってみた。電子ブック内の情報はJISコードで書かれているので（X68000はシフトJISコード），単に変換して表示しているだけだ。本当は漢字IN/OUTコードがあって半角文字も交じっているのだが，今回はちょっと手を抜いて全角文字に変換して表示している。なお，JIS→シフトJIS変換は「試験に出るＸ１」のものをそのまま使わせていただいた。

電子ブックには辞書以外にもさまざまなタイトルが発表されている。なかには特殊なアクセスが必要な場合があるので，こういった単純なダンプツールも役立つことがあるだろう。

電子辞書はけっこう面白い。安いドライブがあったら2，3台増設して辞書を入れっぱなしにしておくのもよいかな，と思ったりもする（普通は多連装ドライブを検討するんだろうな)。

リスト1

```
 10 /*   JISファイルダンププログラム */
 20 int fp,l,j1,j2
 30 char buf(65535)
 40 fp=fopen("m:start","r")
 50 repeat
 60   l=fread(buf,65536,fp)
 70   for i=0 to (l-1)/2
 80     j1=buf(i*2)
 90     j2=buf(i*2+1)
100     if j1=&H1F and j2=&HA then print
110     if j1=&H0  and j2=&H1 then print :print
120     if j1=&H0  and j2=&H0 then continue
130     if j1>&H1F then print sjis(j1,j2) ;
140   next
150 until feof(fp)
160 fclose(fp)
170 end
180 func str sjis(j1,j2)
190 str sj
200 int s1,s2=&H9F
210 if (j1<&H21) or (j1>&H7E) then return("")
220 if (j2<&H21) or (j2>&H7E) then return("")
230 s1=(j1-&H21)/2+&H81
240 if j1>=&H5F then s1=s1+&H40
250 if j1 and 1 then {
260     s2=&H40
270     if &H60<=j2 then s2=s2+1
280 }
290 s2=s2+j2-&H21
300 return(chr$(s1)+chr$(s2))
310 endfunc
```

まずはダンプから

データ解析の勘所

Kikuchi Isao 菊地　功

そのままではまったく使えなくても存在するデータ群
こういったものをうまく解析できれば世界も広がってくるだろう
とりあえず16進ダンプをじっと眺めるところから始めてみよう

いうまでもないことですが，記憶媒体ってのは，なんらかの情報を記録しておくためのものであり，「配布」という一方通行で構わないのであれば，現在もっともコストパフォーマンスに優れたメディアは，やはりCD-ROMということになるでしょう。

で，当然「Windows用」とか「Macintosh用」，あるいは「SATURN用」や「PlayStation用」とかって謳われたCD-ROMソフトは，それらの環境で標準的にサポートされている形式で記録されているか，あるいはその環境で動作するアプリケーションを備えているわけです。

前者としてはCG集であったりムービーであったり，後者としてはワープロソフトであったりゲームであったりするわけです。さてそれでは，それらの環境にないマシンではどうでしょうか。当然のことながら，それらを直接利用することはできません。

しかし，冒頭でも書いたように，基本的に情報の塊であることには間違いないのですから，使い方によってはそれらの情報を有効に活用することはできるはずです。イメージデータやPCMデータ，極端にいえばプログラムだって逆アセンブルして解析すれば，なんらかの情報は得られるでしょう。こういったものの活用に関して，いままでOh!X誌上でも数々取り上げてきました。しかし，ソフトは星の数ほどあるわけで，標準フォーマットならばともかく，ローカルフォーマットまでいちいち解析して，誌面で取り上げるのは不可能です（著作権の問題もありますし）。

なにがいいたいかっていうと，思い立ったら，自分でなんとかしてみようと考えてもらいたいということです。「別にそこまでする必要ないじゃん」って思われた方は，この先は読まなくても結構です（いつになく強気だな）。「解析に懸けるロマン」のわかる方だけに捧げます。

とはいっても，フォーマットの解析というのは，8・9月号のシネパックの記事からもわかっていただけるように，若干の知識と，多少の経験と，多大な熱意（意地ともいう）が必要です。しかし，これらすべてを文章で，しかも数ページに収めることなど不可能です。したがって，ここでは解析に懸ける心構えと，解析のさわりだけを述べることにします。

初めに断っておきますが，この記事を読んだからといって，明日からなんでもばりばり解析できるようになる，というわけではありません。

心構え

まず最初に理解しておかなければならないのは，「解析結果を公表できるとは限らない」ということです。解析結果を迂闊にネットなどで流してしまった場合，それによってどこかから圧力がかかったり，場合によっては訴えられることも十分考えられます。特に最近のアプリケーション（特にPlayStation用）には，リバースエンジニアリング禁止を明言しているものが多くありますので解析自体にも十分な注意が必要です。

この「解析」という作業は，自己満足に浸れる人だけの特権（？）なのかもしれません。

データなどを解析する場合，ある程度そのソフトがターゲットとしているマシンのハードウェアスペックを知っておくと，有効なこともあります。もちろん，そのソフトを動かす環境があれば，一度その動作を確認しておくことも重要です。そうすれば，解像度や色数，PCMの周波数などにあたりをつけることができますからね。

ファイル名から内容を連想することも必要です。はたして，このファイルがイメージデータなのか，PCMデータなのか，あるいは全然別のデータファイルなのか，まったくわからないというのであれば，手のつけようがありませんから。

フォーマットに関しては，「自分ならどうするか」ということを考えましょう。基本的には「いかに無駄を少なくするか」ということですが，場合によっては速度を稼ぐために同じデータが複数箇所に入っていたり，ほかの箇所から求められるデータが入っていたり，時にはサイズを整えるために無意味な詰めものがしてあったりもします（なかにはなにを考えてるのか理解に苦しむのもありますが）。

この辺りはかなり経験がものをいってきますが，とりあえずわからない値は10進数や2進数にしてみます。たとえば，$0140といわれてもピンときませんが，320といわれれば「画像のサイズかな」ってのは容易に思いつきますし，$5622ではなんのことかまずわかりませんが，22050ならば「ああ，PCMの周波数か」ってことになります。

あるいは，絶対もしくは相対アドレスと考えて，そのアドレスを見てみるとか。その辺りでデータのイメージが変わっていたら（これもちょっと経験がいる），その前後のデータはそこで別のものを表していることになるでしょう。

ファイルのサイズなんてのもありがちです。あと，ASCII文字列で識別子を示している場合などもあります。

フォーマットの解析時には，たしかに粘りも必要ですが，あまりひとつのファイルだけにこだわっているのは得策ではありません。同じフォーマットであっても，内容によっては気づきやすいものになっていたりします。

たとえば，込み入った画像よりも，平坦で単純な画像のほうが，普通に考えてもわかりやすそうですよね。同じフォーマットっぽくて，ファイル名などから内容のあたりをつけられる場合は，できるだけ単純なものから手をつけ，行き詰まったら，ほか

のファイルも覗いてみましょう。ここで，なにか些細なことでもわかったら，自分で適当にフォーマットをでっち上げて，試しにローダを作ってしまいます。イメージデータなのであれば，とりあえず横幅を適当に決めて，VRAMに流し込んでみるのも手です。運がよければ，ここできっとなにかが見えてくるでしょう。

いい忘れていましたが，他機種のデータを利用する際には，エンディアンにも注意しなければなりません。エンディアンとは，ワードまたはロングワードのデータがメモリ上にどのようなバイト順で格納されているかを示すもので，高位アドレスに向かってデータの上位バイトから格納されているのをビッグエンディアン，下位バイトから格納されているのをリトルエンディアンといいます。

で，メモリ上でそのように格納されているのですから，それをそのままファイルに落とせば，ファイル中でもそのようなバイト順になってしまいます。基本的にビッグエンディアンかリトルエンディアンかというのはCPUに依存し，Intelのx86などがリトルエンディアン，それ以外のほとんどはビッグエンディアンだと思っていいでしょう(PowerPC, MIPSのRシリーズなどはフラグひとつでどちらにもなる)。X680x0はというと，モトローラの680x0シリーズに依存するので，もちろんビッグエンディアンなわけですから，リトルエンディアンのマシン用のデータを利用するときには，ワード以上のデータの読み込みには工夫が必要になってきます。

イメージデータ

とりあえず，ベタのイメージデータでも拾ってみます。といっても，まずはどれがそのファイルに相当するかというのを見極めなくてはなりません。ここでは，サターン用の「レイアース」を取り上げてみましょう。ベタであれば，ファイルサイズは固定長になるので，同じサイズのファイル，しかもファイル名がそれっぽいものを探します。すると，ENIKKIディレクトリの下にPRESEA00～03.BINというファイルが見つかりました。

ファイルサイズからだいたいの推測もできるのですが，まずはダンプを取ってみましょう(リスト1)。頭のほうは適当にばらけた値が入っていて，これだけではなんのデータかははっきりとはわかりません。しかし，つらつら～っと下のほうに目をやると，$200を過ぎたあたりから，同じバイトデータが連続して現れだします。こういったデータを見つければ，これはべたデータであることはほぼ間違いありません。しかもバイト単位ですから，垂直型の256色だと推測できます。そうなれば，きっとパレットがあるはずですから，$200までがパレットかなぁ，するとカラーコードは$200(512)÷256＝2バイトかなぁってことになってきます。

この時点で$200あたりからをVRAMに流し込んで確信を得てもいいんですが，それ以前はパレットらしいという推測まで立てたのだから，もうちょっと調べてみましょう。ファイルの先頭から4バイトはとりあえず置いといて（きっと識別子かなんかでしょう），4・5バイト目の$0140，「あぁ，さっきなんか出てきたっけ」，そうです，10進にすると320，きっとイメージの横



恥ずかしい写真その1

幅でしょう。とすると，次の$00E0＝224は高さ，ちょうどそれっぽい値じゃないですか。

するってーと，幅320ドット×高さ224ドット×1バイト/ピクセル＋パレット数256×2バイトごとのカラーコード＝72192バイト，このファイルのサイズが72208バイトですから，じゃあ残りは先頭のヘッダか

リスト1

```
00000000  00 01 01 00 01 40 00 E0 00 00 02 10 00 01 18 00 .....@.........
00000010  FF FF 7F FF 7F BD 67 FF 6B BE 73 9C 6F 3F 67 9D .....ｽg.kｾs徙?g揖
00000020  6F 7B 5B 9E 5B 7F 6B 7B 6B 1E 57 7F 5F 9C 6B 5B  {[杣.k{k.W._徂[
00000030  6B 5A 66 FD 53 5E 53 7D 5B 7B 5F 5B 67 39 43 7F kZf.S^S}[{_[g9C.
00000040  4B 7C 53 5B 63 18 4B 5C 5E DB 4A FE 43 5D 47 3C K|S[c.K\^ﾛJ.C]G<
00000050  56 5F 56 7E 4B 1B 33 5F 3F 3D 5A D8 53 18 47 3A V_V~K.3_?=ZﾘS.G:
00000060  42 FD 43 1B 52 3E 2B 3F 52 F7 42 BD 33 1E 42 9E B.C.R>+?R.Bｽ3.B.
00000070  3A FB 3A BD 5A B5 46 F8 3B 1A 27 1E 4E B7 4E 1C :.:ｽZｵF.;.'.NｷN.
00000080  49 FE 36 DA 42 B8 2A FC 4E 95 32 3F 4E 57 36 B9 I.6ﾚBｸ*.N.2?NW6ｹ
00000090  52 73 22 DC 36 1D 45 FA 41 DC 32 7A 32 1D 41 DB Rs"ﾜ6.E.Aﾜ2z2.Aﾛ
000000A0  46 18 32 98 46 74 31 FD 4A 53 32 59 3D 9C 1A 3F F.2炉t1.JS2Y=..?
000000B0  3E 74 2D DD 22 5B 41 D7 22 5A 2E 38 22 3B 46 32 >t-ﾝ"[Aﾗ"Z.8";F2
000000C0  22 3A 29 BC 16 1E 46 31 3A 33 22 1A 0E 7C 2E 55 ":)ｼ..F1:3"..|.U
000000D0  3E 32 2E 17 29 9C 31 F7 3D B6 1E 1A 36 33 29 9B >2..).1.=ｶ..63).
000000E0  29 D9 21 FA 1A 1A 1D F9 36 12 2A 15 12 59 19 F9 )ﾙ!.....6.*..Y..
000000F0  3D F0 1E 36 11 DC 21 5C 31 39 2A 13 25 98 02 3B =  6.ﾜ!\19*.%..;
   :
00000200  08 68 08 86 00 48 08 64 00 65 08 43 04 22 00 00 .h...H.d.e.C."..
00000210  89 89 89 89 89 89 89 7B 89 7B 89 98 9F 89 89 8E 演演演閲閲汚辻猿
00000220  89 89 89 89 8E 89 89 89 89 89 8E 8E 98 8E 98 9F 演演脂演演試 弐
00000230  8E 89 89 89 89 8E 8E 98 98 98 9F B9 B9 BF B9 7B 脂演猿侍  滝ｹｿｹ{
00000240  7B 7B 7B 7B 89 9F 8E 8E 8E 7B 89 89 8E 98 9F 98 {{{{押試施演侍沽
00000250  89 89 8E 89 7B 89 98 7B 89 7B 7B 7B 89 8E 98 89 演脂{汚{閲{{猿
00000260  89 89 8E 8E B9 9F 89 8E 8E 8E 89 89 7B 7B 7B 7B 演試ｹ辻試脂閲{{{
00000270  7B 7B 7B 7B 7B 78 7B 89 89 8E 8E 89 89 7B 7B 89 {{{{{x{演試演{{演
00000280  89 8E 98 8E 8E 9F 9F 9F 98 8E 7B 8E 9F 98 9F 8E  侍試洄沽施次弐試
00000290  8E 89 89 98 98 89 89 8E 98 89 9F 9F 8E 89 89 89  演  演侍押沁演猿
000002A0  8E 9F 98 89 89 98 9F 89 89 98 98 98 8E 89 89 89  沽演弐演    演汚
000002B0  98 9F 9F 98 9F 9F DC 16 01 01 01 01 01 01 10 D9  洄弐洒.......ﾙ
000002C0  CA 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B ﾊ屁屁屁屁屁屁屁
000002D0  9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B  屁屁屁屁屁屁屁屁
000002E0  9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B 9B  屁屁屁屁屁屁屁屁
000002F0  9B 9B 9B 9B 9B 9B 9B 9B D9 01 01 01 01 01 08 CB  屁屁屁巍.....ﾋ
   :
```

な，ってことになります。すると，ヘッダが16バイトでパレットが512バイトだから，イメージは528($210)バイト目からかなって思ったら，高さの次の4バイトがドンピシャリ，そのアドレスを示しているじゃないですか。

それじゃあ，次の4バイトはどうでしょう。$11800，ちょっと大きな値ですが，ちょっとでも経験のある方なら，すぐに思いつくでしょう。その前がイメージ本体の先頭アドレスなら，あと残るはイメージ本体のサイズくらいですよね。実際に計算してみれば，320×224=71680($11800)で，ぴったり大当たりです。

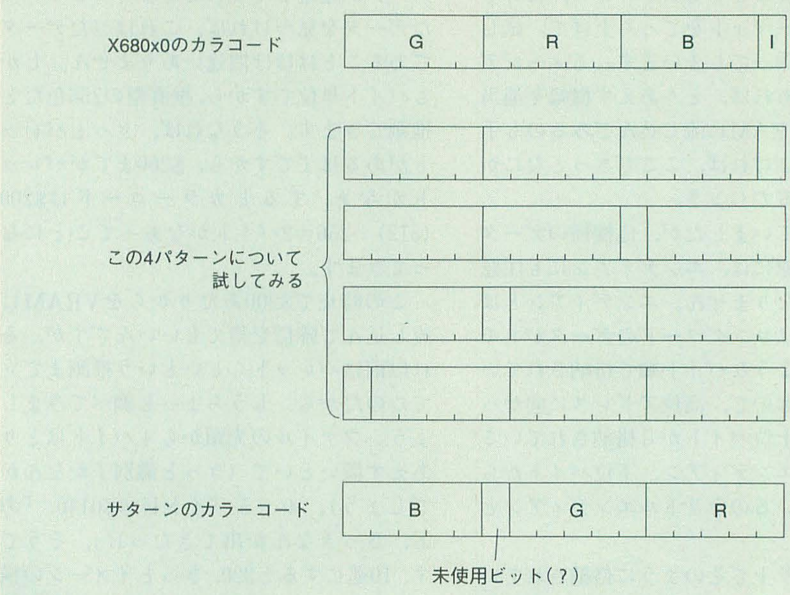
さて，パレットなんですが，カラーコードが16ビットなので64K色だということはわかるのですが，必ずしもX680x0のカラーコードと一致するとは限りません。というより，まず同じであるはずはないでしょう。しかし，おそらくRGB各5ビットで，残りの1ビットはX680x0の輝度ビットのように，あまり気にしなくてよい存在だろうと思われます。

そこで，それらのビット順を調べなければならないのですが，このへんはカラーコードを2進に展開して，赤がどれくらいで……なんて考えるよりも，実際に表示してみるのがいちばんです。RGBそれぞれの5ビットがカラーコード中で分割されているとはちょっと考えにくいので，要は未使用の1ビットがどこにあるかということです。つまり，図1中で示した4パターンについて試してみればいいわけです。

RGBが必ずしもこの順ではありませんが，未使用ビットの位置があっていれば，それなりにちゃんと見えるものです。未使用ビットの位置を調べてから，RGBを適当に入れ替えてみましょう。調べた結果，サターンの16ビットカラーコードは図1下のようになっていました。ちょっと余談ですが，なんでSATURNのカラーコードが16ビットなんでしょうね。パレットはそのマシンの最大ビットカウントのカラーコードで表すのが普通だと思うんですけど。そういやあ，64Kベタのイメージデータは見たことあるけど，フルカラーのは見たことないなぁ。シネパックはYCCだったし。すると，実はサターンって，ハイカラーまでしか出ないのかな？

さて，カラーコードを調べる時点で，すでにローダは完成しているはずですね。一応，リスト2にプログラムを掲載しておきますので，アミュレットのない人でプレセアの昔の写真を見たい人は使ってみてくだ

図1 カラーコードの調査



リスト2

```
#include        <stdlib.h>
#include        <stdio.h>
#include        <doslib.h>
#include        <iocslib.h>

void    main( int ac, char *av[] )
{
        FILE      *fp;
        int       dx, dy, x, y;
        int       r, g, b;
        unsigned int      offset, size;
        unsigned short   col, *vp;

        if( ac<2 ){
                printf( "ファイル名を指定してください。¥n" );
                return;
        }
        fp = fopen( av[1], "rb" );
        if( fp==NULL ){
                printf( "ファイルがありません。¥n" );
                return;
        }
        if( getl( fp )!=0x00010100 ){
                printf( "フォーマットが違うようです。¥n" );
                return;
        }

        dx = getw( fp );          /* イメージ幅 */
        dy = getw( fp );          /* イメージ高さ */
        offset = getl( fp );      /* イメージ本体の先頭アドレス */
        size = getl( fp );        /* イメージデータ本体のサイズ */
        CRTMOD( 8 );
        G_CLR_ON();
        SUPER( 0 );
        vp = (unsigned short *)0xE82000;          /* パレット */
        for( x=16; x<offset; x+=2 ){              /* ヘッダの直後からイメージ本体の直前まで */
                col = getw( fp );
                r = (col)&0x1f;
                g = (col>>5)&0x1f;
                b = col>>11;
                *(vp++) = ((((g<<5)|r)<<5)|b)<<1;
        }
        fseek( fp, offset, 0 );                   /* イメージデータの先頭 */
        vp = (unsigned short *)0xC00000;          /* GRAM */
        for( y=0; y<dy; y++ ){
                for( x=0; x<dx; x++ ) *(vp++) = fgetc( fp );
                vp += 512-dx;
        }
        fclose( fp );
}
```

さい。

PCMデータ

こちらも持っていない方には申しわけないのですが，サターンを中心に話を進めていきます。PCMデータの場合は，ありがたいことにたいていの場合は拡張子が“PCM”になっていますので，どれがPCMデータか迷うことはあまりありません。しかしなかには拡張子が“CPK”で，実はPCMしか入っていないものもあります。シネパックプレーヤーに「未対応のフォーマットです」と怒られたものは，おそらくこれに当てはまります。

PCMデータかどうかというのは，ファイルの先頭を覗いてみれば，たいていわかります(リスト3)。FF,00,01といった値がばらまかれていますね。ここは本来は無音だったのが，データ作成時にノイズが入ってしまった部分で，このような模様があった場合には (少なくともその部分は) PCMデータだと思って間違いありません。ただしこれは，符号ありのPCMの場合の話で，マシンによっては，符号なしだったり符号のつけ方が違っていたりする場合があるようです。

また，周波数が十分高く，バックに「ザー」といったノイズが入っていたりするのでなければ，PCMデータもある程度の連続した値が続きますので，無音部分がないPCMでもその辺りで判断できるかもしれません。

問題は，PCMデータはほとんどが生データで，ヘッダなどもついていないため，周波数やビット数を機械的に判断できないというところです。ビット数に関しては，ダンプリストを眺めて，データが8ビット単位か16ビット単位かで判断することもできますが(リスト3は8ビット)，周波数や，ステレオ/モノラルといったことに関しては，実際に再生して聞いてみないことにはわかりません。聞いたところで，たとえば，22.05kHzステレオか，44.1kHzモノラルかを判断するのは，LRがかなり偏っている場合を除いて，それほど容易ではないでしょう。ステレオのデータをモノラルで再生した場合は，多かれ少なかれノイズが入るはずですが，その辺りは自分の耳を信じるしかありませんね。

いまさらいうまでもないことですが，X680x0はAD PCMですので，これらのPCMを直接再生することはできません。迂闊にPCMデバイスにコピーでもしようものなら，ノイズの嵐でスピーカーを傷めてしまいます。PCMデータをAD PCMに変換して再生するツールなどは，ネットに行けば山ほど転がっているのでしょうが，そんなもの持ってないし通信もしてないという方は，8月号の付録ディスク1，シネパックプレーヤーのソースのadpcm.sにちょっと手を加えるだけで，お手軽にPCMのプレーヤーを作ることができます。

これには，8ビットまたは16ビットのモノラル/ステレオのPCMを15.6kHzのAD PCMに変換する関数が収められていますが，ステレオデータは格納順がちょっと異なっていますので，148行の，

```
move.b  (a4)+,d7
```

を，

```
move.b  (a0)+,d7
```

に，210行の，

```
move.w  (a4)+,d7
```

を，

```
move.w  (a0)+,d7
```

に直してください。

それぞれの関数の仕様については，ソース中に記してありますのでそちらを見ていただきたいのですが，これらの関数は最初にinitPCMtoADPCM()関数を呼んで，あとは細切れになったPCMを次々と放りこめば，ワンパスで次々と繋がったAD PCMデータを生成しますので，大きなPCMの再生にもそれほどメモリを消費しないプログラムを作ることができます。手順を示せば，リスト4のようになるでしょうか。これは完全なプログラムではありませんので，あとは各自でやってみてください。

最後に

今回は解析といいつつ，ベタのイメージデータと，非圧縮のPCMデータしか扱いませんでした。確かにCD-ROMくらいの大容量ならば，多少データが大きくてもびくともしないので，サターンのソフトを見ていてもベタデータが結構たくさんあったのですが，それでも読み込みの時間を短くするためには小さいほうがいいですし，大量のアニメーションデータに圧迫されたりして，なんらかの圧縮がかかっているものも数多くありました。

PCMはサターンに関しては非圧縮なのですが，3DOでは圧縮がかかっている場合もあるようです。しかし，冒頭にも書いたように，今回はあくまでも解析の精神を広めるだけであって，実際に解析を行うのは読者の方々であってほしいという願いを込めた記事だったわけです。あとは皆さんが各自で多大な意地を貫き通してください。その際に，この記事が少しでも参考になれば幸いです。

余談ですが，放流した1匹が帰ってきました。おお，なかなか速くなってるじゃーありませんか。あれ以上はそれほど速くはならないと思ってたのに，ほとんど遅れないなぁ。それとも間引いてるのかな？　しかし，随分太ったなぁ。20倍以上だ。かたっぱしからループ展開したのかな。結構危険なほどに割り込みも使ってるようだけど，ソースがないのでわかんないや。

それにしても，最近こんなことばっかりやってる気がするなぁ。そのうちどっかから鉄槌が飛んできそうだ。ま，飛んでくるまでいいか。

リスト3

```
00000000  00 00 FF 00 FF 00 FF 00 FF 00 00 00 FF 00 FF 00 ................
00000010  FF 00 FF 00 FF 00 FF 00 FF FF 00 FF 00 FF 00 FF ................
00000020  00 FF 00 00 00 FF 00 FF 00 FF 00 FF 00 00 FF FF ................
00000030  00 FF FF 00 FF FF 00 FF FF 00 FF 00 FF 00 00 FF ................
00000040  00 00 FF 00 FE 00 FF 00 FF 00 00 FF 00 FF 00 00 ................
00000050  FF 00 FF 00 FF 01 FE 01 FE 01 FF FF 00 FF 00 FF ................
00000060  FF 00 FF 00 FF 01 FF 00 FF 00 00 FF 00 FF 00 FF ................
00000070  00 FF 00 FF FF 00 FF 01 FF FF 00 FE 01 FD 02 FD ................
00000080  01 FF 01 FE 01 FE 00 00 FF 00 FF FF 00 FF FF 00 ................
00000090  FF 00 FF 00 FF 00 FE 01 FE 00 00 FE 01 FE 01 FF ................
000000A0  FE 01 FE 01 FE 00 00 FF FF 01 FC 02 FD 01 FF FF ................
000000B0  00 FE 00 FF FF FF 01 FE 00 00 FF FF 00 00 FF 00 ................
000000C0  00 FF 00 FF 00 FE 01 FE 01 FF 00 00 FE 01 FE 00 ................
000000D0  00 FE 01 FD 02 FE 01 FE 00 00 FF 00 FE 01 FF 00 ................
000000E0  FF 00 FE 01 FE 01 FE 00 FF FF 01 FE 01 00 FE 01 ................
000000F0  FE 02 FD 02 FE 01 FF 00 00 FF FF 01 FD 03 FC 03 ................
     :
```

リスト4

```
char    PCMBUF[PCMBUFSIZE];
char    ADPCMBUF[ADPCMBUFSIZE];
char    *adpcm;
int     size;
int     freq;   /* 周波数 */
FILE    *fp;    /* PCMファイルのファイルポインタ */

initPCMtoADPCM();
adpcm = ADPCMBUF;
do {
        size = fread( PCMBUF, 1, PCMBUFSIZE, fp );
        adpcm = PCM??toADPCM( PCMBUF, size, adpcm, freq );
} while( size );
ADPCMOUT( ADPCMBUF, 4*256+3, adpcm-ADPCMBUF );
```

BMPローダ2種

BMPファイルを活用する

Kikuchi Isao 菊地 功

巷のCD-ROMでいちばん利用しやすいデータ，BMPファイル
今回はコマンドライン上での一歩進んだ有効利用法を考えよう
640×480ドットサイズビュア用と高品質ビュア用ライブラリだ

今年に入ってから，CD-ROMの特集がありーの，CD-ROMドライブの購買意欲をそそるツールありーので，洗脳されてふらふらとCD-ROMドライブを買ってしまった方も多いことでしょう。

CD-ROMというメディアは，リードオンリーとはいうものの，もうすでにソフトウェアの配布媒体としては確固たる地位を築いています。600Mバイトを超える容量を持ちながら，メディアは1枚数百円（プレスで大量に作った場合），ドライブも4倍速で2万円を割っているとなれば（ベアドライブの秋葉原価格），まだ当分はその地位も安泰というものです。問題があるとすれば読み込みが遅いということですが，6倍速もすでに3万円台，いずれ8倍速，10倍速というものも出てくるでしょう。

さらに，CD-Rドライブもかなりの値下がりを見せてきていますので，個人でCD-ROMを焼いて，コミケなどで配布されることも多くなるはずです。それを考えると，「68ではいまいち有用性ないから……」なんていってる場合じゃありません。そんなものは買ってから考えればいいのです。さあ，いますぐショップに走りましょう。

さあ，買ってきましたね？ え？ サターンのソフトも一緒に買ってきた？ なんだ，ちゃんと考えてるじゃないですか。って，それはちょっとこっちに置いておいて，もう少し別の利用法を考えてみます。X68000用のCD-ROMとしては，先月に発売されたSX-WINDOW開発キットはCD-ROM付きですし，もうすぐ発売される（はずの）EX-SystemにもCD-ROMがつく予定です。

が，世の中はMS-WindowsとMacintosh一色（いや二色か）なわけで，当然そういったマシン専用のCD-ROMが圧倒的なのはいうまでもありません。そこで，いまは亡き（?）石上氏が1月号でもやりましたが，MS-Windowsのグラフィックの標準フォーマットであるBMPについて，もうちょっといじくりまわしてみましょう。あ，今回はSXじゃありませんので，「SXは使ってないよ～」という方もちょっと安心です（私もそうだから）。

BMPのフォーマットについては，1月号の記事を参照しました。BMPにはローカルフォーマットが大量に存在するようで「あれだけじゃ全然足ら～ん！」という意見もあったようですが，私は特に困っていないので無視することにします。あくまでも普通のCD-ROMに入っているグラフィックデータが相手ですから。

ただし，手元にあったBMPのうち，必要なデータが0で埋め尽くされているものがあったので，そのへんは適当になんとかなるようにしてあります（なんて曖昧な）。また，1月号同様に，非圧縮のみの対応となっています（OS/2には対応しています）。

BMPLOADライブラリ

とりあえず，BMPファイルをロードするためのライブラリを作ってみました（リスト1～6）。リスト中にコメントで説明が入っていますが，一応説明しておきましょう。

```
int GetBMPHeader(
        FILE            *fp,
        BMPHEADER       *bmp
    );
```

fpで指定されたファイルをBMPファイルと見なし，情報をbmpが示すBMPHEADER構造体に格納します。ライブラリ中の他の関数を使用する前に，必ず呼び出す必要があります。戻り値が0で正常終了，－1であれば，対応していない形式であることを示します。BMPHEADER構造体については，BMPLOAD.Hで定義してあります。関数をコールしたあとに必要であれば参照してください。

```
int GetBMPPalet(
        FILE            *fp,
        BMPHEADER       *bmp,
        unsigned char   *pal
    );
```

fpで指定されたBMPファイルから，bmpが示すBMPHEADER構造体の情報を基にパレットデータをpalが示す領域に取り出します。カラーコードは24ビットデータですので，少なくともBMPHEADER内の情報ClrUsed×3バイトの領域を確保しておかなければなりません。パレットデータは，各8ビットでB1,G1,R1,B2,G2,R2,……の順で格納されます。戻り値は読み込んだパレットの数で，パレットのないBMPを指定した場合は0を返します。

```
int GetBMPImage(
        FILE            *fp,
        BMPHEADER       *bmp,
        unsigned char   *buf
    );
```

fpで指定されたBMPファイルから，bmpが示すBMPHEADER構造体の情報を基にイメージデータをbufが示す領域に取り出します。データバイトがラインをまたぐことはできませんので，少なくとも（Width×BitCount÷8以上の整数）×Heightバイトの領域を確保しておかなければなりません。BitCountが4以下（16色以下）の場合は，バイトデータの上位から順に，24（フルカラー）の場合は，各8ビットでB,G,Rの順で格納されます。

注意が必要なのは，ラインデータ内のピクセルデータは画面左から右順で格納されるのですが，ラインデータそのものは画面下から上へ格納されます。これはBMPそのものがそのようにデータを格納しているせいで，おそらく数学で一般的に使われる座標軸の向きと一致させたいという思惑があったのでしょう。ちょっと迷惑な話です。

```
int GetBMPLineFirst(
        FILE            *fp,
```

```
        BMPHEADER       *bmp,
        unsigned char   *buf,
        BMPLINEBUF      *lb
    );
```

fpで指定されたBMPファイルのlbが示すBMPLINEBUF構造体のy（いちばん上を0，下向きが正として数えたライン番号）で示されたラインデータをbufが示す領域に取り出し，BMPLINEBUF構造体を更新します。少なくともWidth×BitCount÷8以上の整数バイトの領域を確保しておかなければなりません。

続けてGetBMPLineNext()をコールすれば連続してラインデータを読むことができますが，ライン番号は負方向に進みます。また，これらの関数を呼ぶ間でファイルをシークしないでください。BMPLINEBUF構造体についてはBMPLOAD.Hで定義していますが，ユーザープログラムで知っておく必要があるのはライン番号だけです。

```
int GetBMPLineNext(
        FILE            *fp,
        unsigned char   *buf,
        BMPLINEBUF      *lb
    )
```

fpで指定されたBMPファイルからlbが示すBMPLINEBUF構造体を参照して，ラインデータを取り出し，BMPLINEBUF構造体を更新します。この関数をコールする前に必ずGetBMPLineFirst()関数を呼び出してください。

これらの関数の使用手順としては，

```
         GetBMPHeader()
               ↓
         GetBMPPalet()
          ↓          ↓
GetBMPImage()  GetBMPLineFirst()
                     ↓  ←─┐
             GetBMPLineNext()─┘
```

というのが普通でしょう。一度にメモリにすべてのデータを持つ必要がない場合は，GetBMPLineのほうを利用して，メモリの節約を図るべきです。

なお，このライブラリの構築は，今回のプログラムとまとめてリスト11のメイクファイルで行うことができます。以降のプログラムは，このライブラリを使用しますので必ず入力してください。

256色専用ローダ

768×512ドットの画面モードで，512×512ドットの範囲内に限ってハイカラーを表示する方法は，さまざまなツールで使用されていますので，かなり知れ渡っていることでしょう。しかし，768×512ドット画面全体で256色表示する方法はあまり知られていないようです。というのも，この画面モードを作り出すのにCPUパワーをほとんど喰い潰されてしまい，表示する以外にほとんどなにもできなくなるでしょうから。

この画面モードは特定の利用環境にはかなり重要な意味があります。たとえば，MS-Windowsのアプリケーションなどでは，640×480ドット256色というのをローエンドに設定してあるものがほとんどです。当然，巷で出まわっているBMPファイルにしても同じことがいえます。もうおわかりですね。この解像度がすっぽり収まってしまう画面の大きさというのは，こういったデータを表示する場合には大変有効なわけです。

では768×512ドット256色モードにするにはどうしたらいいのでしょう。普通に256色モードにすると，512×512ドットのプレーンが2枚できます。この2枚のプレーンには，表示したい768×512ドット画像のそれぞれ左右512ドットを描画し，右側を描画したプレーンは256ドットスクロールさせておきます。

さて，この状態で768×512ドットモードにしても，図1左のようになるだけで，正しく表示されません。しかし，この2つのプレーンを真ん中ですぱっと切って，繋ぎあわせればちょうどぴったりあうことがおわかりになるでしょう。

さて，ここで考えます。モニターに表示されている画面っていうのは，なにもVRAMを一度に表示しているのではなく，左上から右方向に，いちばん右までいったら一段下の左端からまた右方向にと，順に表示されていっているというのはいまさらいうまでもないでしょう。これを利用したラスタスクロールという手法は，よくご存じのはずです。

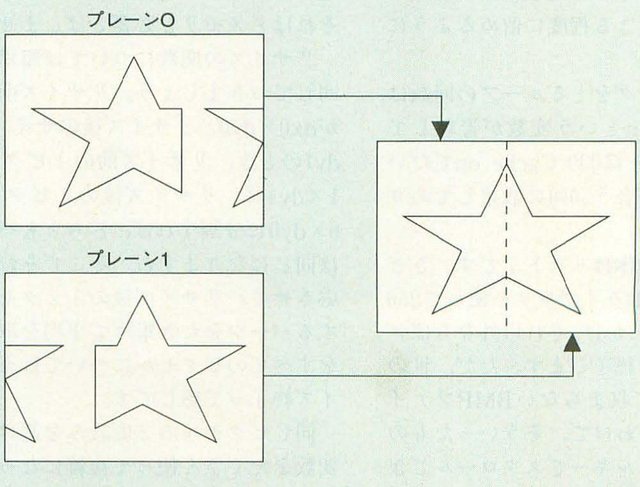
では，それと同じように，走査線が画面左半分にあるときには一方のプレーンだけを，右半分のときはもう片方のプレーンだけを表示するようにすれば，繋がって見えるはずです。

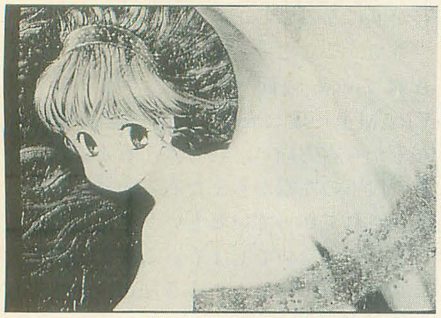
ただ違うのは，ラスタスクロールは画面を横に切るのに対して，縦に切らなければならないということです。

些細なことのように思われるかもしれませんが，実はこれは大変大きな問題です。だって，ラスタスクロールは画面を1回「切る」のに，1回の画面書き換え（約1/60秒）に1回の割合の操作しか必要ありませんが，縦に「切る」には，ラスタを1本書く（約1/31ミリ秒）たびにレジスタを操作しなければならないとなれば，その忙しさは比べものになりません（俗に縦ラスタスクロールと呼ばれる手法を使っているゲームがまれにあるが，全然別の手法のはず）。

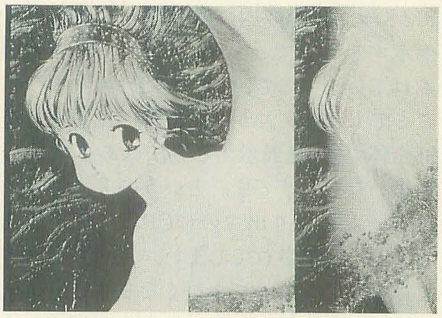
それでもたかだか31kHzと嵩を括って，CPUの負荷を減らすためにCRTCの操作をDMAにやらせてみたり，H-SYNC割り込みやTIMER-D割り込みを駆使してみたのですが，なかなかどうして，タイミング

図1　2プレーンの合成





同期をとって全画面表示



同期があわないとこうなる

がシビアすぎて，ことごとく失敗してしまいました。

結局表示期間はH-SYNCをずっと監視し，ラスタの表示開始からdbraでタイミングを取ってCRTCを操作するという，もっとも安易な方法となってしまいました。しかし，このタイミングのループが10MHzマシンではほんの数回であることを考えると(30 Cache onでは40〜50回)，やむをえないことであることがわかっていただけるかと思います。しかも，OSから割り込みがかかるだけで画面が乱れてしまうので，割り込みもNMI以外すべて禁止してあります。

結果，なにかできるのは垂直帰線期間の，時間的には約1割だけとなってしまいました。まあ，しかたないでしょう。ここは諦めるしかないようです。

この画面モードを作り出すのがリスト8のPageSync()関数です。この関数は，呼び出されると1回の表示期間だけ上の方法で768×512ドットで256色を表示して帰ってきますので，この画面モードを保ちたい場合は，ループの中などに入れて連続して呼ばなければなりません。呼び出しが遅れると，画面の上から画像が乱れてしまいますので，ループ中のほかの処理は必ず垂直帰線期間内に処理できる程度に留めるようにしてください。

また，タイミングをとるループの回数は，同リスト内のWaitという変数が管理しています(デフォルトは030 Cache onでだいたいタイミングが合う50回に設定してあります)。

プログラムの本体はリスト7です。さきほどのBMPLOADライブラリを使って256色BMPをふるいにかけ，それ以外ならばメッセージを出して終了します。ただ，世の中には768×512で収まらないBMPファイルもたくさんあるわけで，そういったもののために，カーソルキーでスクロールできるようにしておきました(すべてメモリに持ちますので，それなりにたくさんのメモリを必要とします)。スクロール中は画面が壊れてしまいますが，勘弁してください。

画面切り替えのタイミングがあっていない場合は，ROLL UP，DOWNキーでウェイトポインタを調整できます。一度調整すると，BMP256.DATというファイルが作られ，それ以降は調整の必要がなくなります。ESCキーで終了します。

同様な手法での表示ツールには画面の切り替え時に画面中央に少しノイズが出てしまうものもありますが，これは画面切り替え時に一瞬両ページ表示を行って前の画面を消せばノイズを消すことができます。

リサイズローダ

2つ目は，任意のサイズ，任意の色数のBMPファイルを，512×512に収まるようにリサイズして，ハイカラーでVRAMに表示します(リスト9)。リサイズは，BMPファイルのアスペクト比が1:1，VRAMが3:4として行い，結構真面目に平均を取っているので，そこそこ綺麗にいくはずです。ただし，BMPファイルのサイズにもよりますが，それなりに時間がかかるのは覚悟しておいてください。BMPファイルからの読み込みは，ライン単位で行っていますので，それほどメモリを必要とはしません。

リサイズの関数については簡単にだけ説明しておきましょう。リサイズ前のサイズがdx0×dy0，リサイズ後のサイズがdx1×dy1のとき，リサイズ前の1ピクセルをdx1×dy1に，リサイズ後の1ピクセルをdx0×dy0に分割すれば，どちらもパーツの数は同じになりますね。そこでそれぞれを対応させて，リサイズ後の1ピクセルに相当するパーツをかき集めて平均を取り，それをすべてのピクセルについて行えば，リサイズ終了って感じです。

同じピクセルの2度読みを避けるために，変数をたくさん使って複雑になっていますが，基本は変わりません。なんで2度読みを避けてるかって？　そりゃ，美的感覚の問題です。ちなみに，フルカラーからハイカラーへの減色は，VRAMに書き出す直前に行っています。あと，減色時にディザはいらないって人は，コメント以降の3行を削除してください。

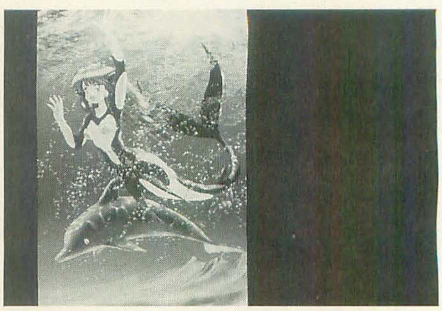
GLMコンバータ

あとなにかやりたかったらMATIERでやってってな感じでなんか投げやりっぽいですけど，最後はGLMへのコンバータです（リスト10)。ピクセル数やアスペクト比はそのままコンバートしますが，GLMなんでBMPの色数がいくつであっても64K色になります。ただし，オプションで領域を指定すると，BMPファイルのその領域を抜き出してGLMに変換します。また，GLMのファイル名を省略した場合は，BMPファイル名の拡張子を"GLM"と置き換えたファイル名にコンバートします。

ファイルからファイルへの転送なんて，できるだけまとめて読み書きしたいのですが，BMPは下から，GLMは上から格納しなければならないし，メモリが少ない場合も考慮しなければなりません。そこで，メモリをできるだけ確保して，それを単位に書き出しています。たとえば，10ライン分のメモリが確保できたのなら，BMPからはまず9から0ラインまでのデータを読み込んで64K色に変換し（変換の時点でラインの順を入れ替える)，10ラインをまとめてGLMに掃き出すといった具合です。このようにしないと，1ラインごとにファイルをシークしなければならなくなりますからね。

コンパイル

ライブラリのところでも書きましたが，コンパイルにはリスト11のMakefileを使います。リストを入力したら，すべてのファイルを同じディレクトリに置き，Make一発全自動でできあがってるはずです。アセンブラとコンパイラが手持ちのものと違っている場合は，メイクファイルを各自の環境に合わせて書き換えてくださいね。あ



できるだけ綺麗に，できるだけ大きく表示

と，ひょっとしたらGNUのMakeではだめかもしれませんので，その辺は各自でなんとかしてください。

調べてないんですけど（あんまり調べる気もしないけど），WarpやWindows95でもBMPのフォーマットって同じなんですかね。

いまさらWindows95に期待するものなんかないけど，きっと'96年は95一色なんだろうな。編集のU氏にはAT互換機を買わせちゃったけど（これはCD-Rのため），Pentium-120ならその次くらいまではもつかな。P6はイマイチみたいだし。

リスト1

```
/* BMP ロード関数           (c)1995 Isawo-Kikuchi    */

typedef struct BMPHEADER {
    int      OffBits;       /* イメージデータまでのオフセット */
    int      Size;          /* Information Headerのサイズ */
    int      Width;         /* 画像データ幅 */
    int      Height;        /* 画像データ高さ */
    short    BitCount;      /* ビット毎ピクセル */
    int      LineBytes;     /* ファイル中でのバイト毎ライン */
    int      ClrUsed;       /* パレット数 */
} BMPHEADER;

typedef struct BMPLINEBUF {
    int      y;             /* ライン番号 */
    int      x;             /* 1ラインのバイト数 */
    int      d;             /* 次のラインまでのバイト数 */
} BMPLINEBUF;

int        GetBMPHeader( FILE *, BMPHEADER * );
int        GetBMPPalet( FILE *, BMPHEADER *, unsigned char * );
int        GetBMPImage( FILE *, BMPHEADER *, unsigned char * );
int        GetBMPLineFirst( FILE *, BMPHEADER *, unsigned char *, BMPLINEBUF * );
int        GetBMPLineNext( FILE *, unsigned char *, BMPLINEBUF * );
```

リスト2

```
/* BMP ロード関数  GetBMPHeader    (c)1995 Isawo-Kikuchi    */
#include    <stdlib.h>
#include    <stdio.h>
#include    "bmpload.h"

/* リトルエンディアン読み込みマクロ */
#define     getlong( fp ) (fgetc( fp )|(fgetc( fp )<<8)|(fgetc( fp )<<16)|(fgetc( fp )<<24))
#define     getword( fp ) (fgetc( fp )|(fgetc( fp )<<8))

/* BMPのヘッダを読んで、情報をBMPHEADER構造体に格納する。              */
/* 戻り値  0..正常終了      -1..対応していない形式                     */
int         GetBMPHeader( FILE *fp, BMPHEADER *bmp )
{
    int     ret = -1;

    rewind( fp );
    if( getw( fp )=='BM' ){
            fseek( fp, 8, 1 );
            bmp->OffBits = getlong( fp );
            bmp->Size = getlong( fp );
            switch( bmp->Size ){
            case 12:/* OS/2 1.x */
                    bmp->Width = getword( fp );
                    bmp->Height = getword( fp );
                    if( getword( fp )!=1 ) break;
                    bmp->BitCount = getword( fp );
                    bmp->LineBytes = 0;
                    bmp->ClrUsed = 1<<bmp->BitCount;
                    ret = 0;
                    break;
            case 40:/* Windows 3.x */
                    bmp->Width = getlong( fp );
                    bmp->Height = getlong( fp );
                    if( getword( fp )!=1 ) break;
                    bmp->BitCount = getword( fp );
                    if( getlong( fp ) ) break;
                    bmp->LineBytes = getlong( fp )/bmp->Height;
                    fseek( fp, 8, 1 );
                    bmp->ClrUsed = getlong( fp );
                    ret = 0;
                    break;
            case 64:/* OS/2 2.x */
                    bmp->Width = getword( fp );
                    bmp->Height = getword( fp );
                    if( getword( fp )!=1 ) break;
                    bmp->BitCount = getword( fp );
                    if( getlong( fp ) ) break;
                    bmp->LineBytes = getlong( fp )/bmp->Height;
                    fseek( fp, 8, 1 );
                    bmp->ClrUsed = getlong( fp );
                    ret = 0;
                    break;
            }
            if( bmp->LineBytes==0 ){
                    bmp->LineBytes = (bmp->Width+3)&0xfffffffc;
                    bmp->LineBytes = bmp->LineBytes*bmp->BitCount/8;
            }
            if( bmp->ClrUsed==0 ) bmp->ClrUsed = (bmp->OffBits-bmp->Size-14)/4;
    }
    return( ret );
}
```

リスト3

```
/* BMP ロード関数  GetBMPPalet      (c)1995 Isawo-Kikuchi    */
#include    <stdlib.h>
#include    <stdio.h>
#include    "bmpload.h"

/* BMPHEADER構造体の情報を参照して、BMPファイルからパレットを読み込む。    */
/* パレットは各8ビットB1,G1,R1,B2,G2,R2...の順                          */
/* 戻り値  読み込んだパレット数                                          */
int         GetBMPPalet( FILE *fp, BMPHEADER *bmp, unsigned char *pal )
{
    int     i, ret = 0;

    fseek( fp, bmp->Size+14, 0 );
    switch( bmp->Size ){
    case 12:/* OS/2 1.x */
            ret = fread( pal, 3, bmp->ClrUsed, fp );
            break;
    default:/* Windows 3.x OS/2 2/x */
            for( i=0; i<bmp->ClrUsed; i++, pal+=3 ){
                    ret += fread( pal, 3, 1, fp );
                    fseek( fp, 1, 1 );
            }
            break;
    }
    return( ret );
}
```

リスト4

```
/* BMP ロード関数  GetBMPImage      (c)1995 Isawo-Kikuchi    */
#include    <stdlib.h>
#include    <stdio.h>
#include    "bmpload.h"

/* BMPHEADER構造体の情報を参照して、BMPファイルからイメージを読み込む。   */
/* 16色以下の場合は、1ラインが画像幅を越える整数バイト分用意すること。    */
/* 戻り値  0..正常終了      -1..エラー                                  */
int         GetBMPImage( FILE *fp, BMPHEADER *bmp, unsigned char *buf )
{
    int     x, y, d;

    x = (bmp->Width*bmp->BitCount+7)/8;
    d = bmp->LineBytes-x;
    fseek( fp, bmp->OffBits, 0 );
    for( y=bmp->Height-1; y>=0; y-- ){
            if( fread( &buf[x*y], 1, x, fp )<x ) break;
            if( d ) fseek( fp, d, 1 );
    }
    return( y?-1:0 );
}
```

リスト5

```
/* BMP ロード関数  GetBMPLineFirst (c)1995 Isawo-Kikuchi    */
#include    <stdlib.h>
#include    <stdio.h>
#include    "bmpload.h"

// BMPHEADER構造体の情報を参照して、BMPファイルのBMPLINEBUF構造体で指定
// されたラインを読み込む（BMPLINEBUF構造体はライン番号のみを指定する）。
// 連続してラインを読む場合は、続けてGetGLMLineNext()をコールするが、
// ラインは負方向に進むので注意すること。
// 戻り値  0..正常終了      -1..エラー
int         GetBMPLineFirst( FILE *fp, BMPHEADER *bmp, unsigned char *buf, BMPLINEBUF *lb
{
    if( lb->y<0 || lb->y>=bmp->Height ) return( -1 );
    lb->x = (bmp->Width*bmp->BitCount+7)/8;
    lb->d = bmp->LineBytes-lb->x;
    fseek( fp, bmp->OffBits+(bmp->Height-lb->y-1)*bmp->LineBytes, 0 );
    if( fread( buf, 1, lb->x, fp )<lb->x ) return( -1 );
    if( lb->d ) fseek( fp, lb->d, 1 );
    lb->y--;
    return( 0 );
}
```

リスト6

```
/* BMP ロード関数  GetBMPLineNext  (c)1995 Isawo-Kikuchi    */
#include    <stdlib.h>
#include    <stdio.h>
#include    "bmpload.h"

// BMPLINEBUF構造体を参照して、BMPファイルの次のラインを読み込む。
// 必ずGetBMPFirst()をコールした後に呼ぶこと。
// これらの関数を呼ぶ間でファイルをシークさせないこと。
// ラインは負方向に進む。
// 戻り値  0..正常終了      -1..エラー
int         GetBMPLineNext( FILE *fp, unsigned char *buf, BMPLINEBUF *lb )
{
    if( lb->y<0 ) return( -1 );
    if( fread( buf, 1, lb->x, fp )<lb->x ) return( -1 );
    if( lb->d ) fseek( fp, lb->d, 1 );
    lb->y--;
    return( 0 );
}
```

リスト7

```
#include   <stdlib.h>
#include   <stdio.h>
#include   <string.h>
#include   <doslib.h>
#include   <iocslib.h>
#include   "bmp.h"

void       Aspect( void );
void       ShowImage( unsigned char *, int, int );

extern void        PageSync( void );
extern short       Wait;

unsigned char      filename[90];
unsigned char      pal[256][3];

void       main( int ac, char *av[] )
{
    FILE    *fp;
    int     i, wait, r, g, b;
    unsigned char   *buf;
    unsigned short  *p;
    struct PDBADR   *pdb;
    BMPHEADER       bmp;

    printf( "256 Color BMP Loader v.1.00¥t(c)1995 Isawo-Kikuchi¥n" );
    if( ac<2 ){
            printf( "[usage] BMP256 <filename>¥n"
                    "¥t256色のBMPファイルを表示します。¥n"
                    "¥t大きな画像はカーソルキーでスクロールすることができます。¥n"
                    "¥tウェイトポインタが合っていない場合は、"
                    "ROLL UP,DOWNで調整してください。¥n"
            );
            return;
    }
    strmfe( filename, av[1], "BMP" );
    fp = fopen( filename, "rb" );
    if( fp==NULL ){
            printf( "ファイルがありません。¥n" );
            return;
    }
    if( GetBMPHeader( fp, &bmp )<0 ){
            printf( "このBMPファイルには対応していません。¥n" );
            return;
    }
    if( bmp.BitCount!=8 ){
            printf( "256色BMPではありません。¥n" );
            return;
    }
    printf( "XSize:%4d YSize:%4d¥n", bmp.Width, bmp.Height );
    GetBMPPalet( fp, &bmp, pal );
    buf = (unsigned char *)MALLOC( bmp.Width*bmp.Height );
    if( buf>0x80000000 ){
            printf( "メモリが足りません。¥n" );
            return;
    }
    printf( "Now Loading.." );
    GetBMPImage( fp, &bmp, buf );
    fclose( fp );
    pdb = GETPDB();
    strcpy( filename, pdb->exe_path );
    strcat( filename, pdb->exe_name );
    strmfe( filename, filename, "DAT" );
    fp = fopen( filename, "rb" );
    if( fp ){
            Wait = getw( fp );
            fclose( fp );
    }
    wait = Wait;
    CRTMOD( 8 );
    G_CLR_ON();
    SUPER( 0 );
    p = (unsigned short *)0xe82000;
    for( i=0; i<bmp.ClrUsed; i++ ){
            b = pal[i][0]>>3;
            g = pal[i][1]>>3;
            r = pal[i][2]>>3;
            *(p++) = ((((g<<5)|r)<<5)|b)<<1;
    }
    OS_CUROF();
    Aspect();
    ShowImage( buf, bmp.Width, bmp.Height );
    MFREE( buf );
    OS_CURON();
    fprintf( stderr, "¥x1B*¥x1B[>1l¥x1B[>5l" );
    if( wait!=Wait ){
            fp = fopen( filename, "wb" );
            if( fp ){
                    putw( Wait, fp );
                    fclose( fp );
                    printf( "ウェイトポインタを更新しました。¥n" );
            }
    }
}

void       Aspect()
{
    unsigned short  *p;

    p = (unsigned short *)0xE80000;
    *(p++) = 0x89;
    *(p++) = 0x0E;
    *(p++) = 0x1C;
    *(p++) = 0x7C;
    p = (unsigned short *)0xE80028;
    *p = 0x0116;
    p = (unsigned short *)0xE82600;
    *p |= 0x20;
}

static struct TXFILLPTR    txfill;

void       ShowImage( unsigned char *p, int dx, int dy )
{
    int     sx1, ex1, sx2, ex2, sy, ey;
    int     x, y, px, py, i, hx, hy, xx, yy;
    unsigned short  *vp1 = (unsigned short *)0xC00000;
    unsigned short  *vp2 = (unsigned short *)0xC80000;

    if( dx<768 ){
            sx1 = (768-dx)/2;
            ex1 = sx1+dx;
            if( ex1>512 ) ex1 = 512;
            sx2 = sx1-256;
            ex2 = sx2+dx;
            if( sx2<0 ) sx2 = 0;
    } else { sx1 = sx2 = 0; ex1 = ex2 = 512; }
    if( dy<512 ){ sy = (512-dy)/2; ey = sy+dy; }
    else { sy = 0; ey = 512; }
    HOME( 1, 0, 0 );
    HOME( 2, 256, 0 );
    TPALET( 0, 1 );
    TPALET( 1, 0 );
    txfill.vram_page = 0;
    txfill.x = sx1;
    txfill.y = sy;
    txfill.x1 = dx;
    txfill.y1 = dy;
    txfill.fill_patn = 0xffff;
    TXFILL( &txfill );
    for( y=sy, py=0; y<ey; y++, py++ ){
            for( x=sx1, px=0; x<ex1; x++, px++ ){
                    vp1[x+y*512] = p[px+py*dx];
            }
            for( x=sx2, px=256-sx1+sx2; x<ex2; x++, px++ ){
                    vp2[x+y*512] = p[px+py*dx];
            }
    }
    hx = hy = 0;
    for ( ;; ){
            i = BITSNS( 7 );
            if( i&8 ){      /* ← */
                    if( hx>0 ){
                            hx--;
                            HOME( 1, hx%512, hy%512 );
                            HOME( 2, (hx+256)%512, hy%512 );
                            x = hx%512;
                            px = hx;
                            py = hy;
                            for( y=sy; y<ey; y++, py++ ){
                                    yy = (y+hy)%512;
                                    vp1[x+yy*512] = p[px+py*dx];
                                    vp2[x+yy*512] = p[px+256+py*dx];
                            }
                            continue;
                    }
            }
            if( i&0x10 ){   /* ↑ */
                    if( hy>0 ){
                            hy--;
                            HOME( 1, hx%512, hy%512 );
                            HOME( 2, (hx+256)%512, hy%512 );
                            y = hy%512;
                            px = hx;
                            py = hy;
                            for( x=sx1; x<ex1; x++, px++ ){
                                    xx = (x+hx)%512;
                                    vp1[xx+y*512] = p[px+py*dx];
                            }
                            px = hx+256-sx1+sx2;
                            for( x=sx2; x<ex2; x++, px++ ){
                                    xx = (x+hx)%512;
                                    vp2[xx+y*512] = p[px+py*dx];
                            }
                            continue;
                    }
            }
            if( i&0x20 ){   /* → */
                    if( hx+768<dx ){
                            hx++;
                            HOME( 1, hx%512, hy%512 );
                            HOME( 2, (hx+256)%512, hy%512 );
                            x = (hx-1)%512;
                            px = 511+hx;
                            py = hy;
                            for( y=sy; y<ey; y++, py++ ){
                                    yy = (y+hy)%512;
                                    vp1[x+yy*512] = p[px+py*dx];
                                    vp2[x+yy*512] = p[px+256+py*dx];
                            }
                            continue;
                    }
            }
            if( i&0x40 ){   /* ↓ */
                    if( hy+512<dy ){
                            hy++;
                            HOME( 1, hx%512, hy%512 );
                            HOME( 2, (hx+256)%512, hy%512 );
                            y = (hy-1)%512;
                            px = hx;
                            py = 511+hy;
                            for( x=sx1; x<ex1; x++, px++ ){
                                    xx = (x+hx)%512;
                                    vp1[xx+y*512] = p[px+py*dx];
                            }
                            px = hx+256-sx1+sx2;
                            for( x=sx2; x<ex2; x++, px++ ){
                                    xx = (x+hx)%512;
                                    vp2[xx+y*512] = p[px+py*dx];
                            }
                            continue;
                    }
            }
            if( i&1 )       /* Roll up */
                    if( Wait>1 ) Wait--;
            if( i&2 )       /* Roll down */
                    if( Wait<100 ) Wait++;
            if( BITSNS(0)&2 ) break;
            PageSync();
    }
    KFLUSHIO( 0xFF );
    TPALET( 0, -2 );
    TPALET( 1, -2 );
}
```

▶『C MAGAZINE』のSX-WINDOWプログラミングが最終回。そのなかで，外付けSCSI-HD内部のスタブ取りつけ方法によって不安定になるという記事が，参考になった。

中野　祥貴(30)長崎県

リスト8

```
        .xdef   _Wait
        .xdef   _PageSync

        .text
        .even

_PageSync:
        movem.l d0-d7/a1-a2,-(sp)

        movea.l #$E82600,a1     * videoc
        movea.l #$E88001,a2     * mfp

        move.w  (a1),d0
        andi.w  #$FFF0,d0
        move.w  d0,d1
        ori.w   #3,d1           * page 0
        move.w  d0,d2
        ori.w   #$C,d2          * page 1
        move.w  d0,d6
        ori.w   #$F,d6          * page 0&1

        move.b  #4,d3           * v-disp bit
        move.b  #7,d4           * h-sync bit

        move.w  sr,d5
        ori.w   #$0700,sr       * 割り込み禁止
        move.w  _Wait,d7
v_wait:
        btst.b  d3,(a2)
        beq     v_wait          * 垂直表示期間を待つ
h_wait00:
        btst.b  d4,(a2)
        beq     h_wait00        * 水平帰線期間を待つ
h_wait0:
        btst.b  d4,(a2)
        bne     h_wait0         * 水平表示期間を待つ

loopy:
        move.w  d7,d0
wait:
        dbra    d0,wait         * 切り替えのタイミングを待つ

        move.w  d6,(a1)
        move.w  d2,(a1)         * page1 に切り替える
h_wait:
        btst.b  d4,(a2)
        beq     h_wait          * 水平帰線期間を待つ

        move.w  d1,(a1)         * page0 に切り替える

        btst.b  d3,(a2)
        beq     quit            * 表示期間が終わっていれば終了
h_wait2:
        btst.b  d4,(a2)
        bne     h_wait2         * 水平表示期間を待つ

        bra     loopy
quit:
        move.w  d5,sr

        movem.l (sp)+,d0-d7/a1-a2
        rts

        .data
        .even

_Wait:  dc.w    50

        .end
```

リスト9

```
#include    <stdlib.h>
#include    <stdio.h>
#include    <iocslib.h>
#include    <doslib.h>
#include    <string.h>
#include    "bmp.h"

void        Resize( int, int, int, int, FILE *, BMPHEADER * );
void        Convert24bit( BMPHEADER * );

unsigned char       filename[90];
unsigned char       pal[256][3];
unsigned char       *buf1;  /* 1ライン生データ */
unsigned char       *buf2;  /* FullColor変換1ラインデータ */

void        main( int ac, char *av[] )
{
    FILE    *fp;
    BMPHEADER       bmp;
    int     x, y;

    printf( "BMP to HighColor512x512 Loader v.1.00¥t(c)1995 Isawo-Kikuchi¥n" );
    if( ac<2 ){
            printf( "[usage] BMPFIT <filename>¥n"
                    "¥tBMPファイルを512x512 64k色に変換して表示します。¥n"
            );
            return;
    }
    strmfe( filename, av[1], "BMP" );
    fp = fopen( filename, "rb" );
    if( fp==NULL ){
            printf( "ファイルがありません。¥n" );
            return;
    }
    if( GetBMPHeader( fp, &bmp )<0 ){
            printf( "このBMPファイルには対応していません。¥n" );
            return;
    }
    x = (bmp.Width*bmp.BitCount+7)/8;
    buf1 = (unsigned char *)MALLOC( x+bmp.Width*3 );
    if( buf1>0x81000000 ){
            printf( "メモリが確保できません。¥n" );
            return;
    }
    buf2 = (unsigned char *)&buf1[x];
    GetBMPPalet( fp, &bmp, pal );
    if( bmp.Width>bmp.Height*4/3 ){
            x = 512;
            y = 512*bmp.Height*4/3/bmp.Width;
    } else {
            x = 512*bmp.Width*3/4/bmp.Height;
            y = 512;
    }
    CRTMOD( 12 );
    G_CLR_ON();
    SUPER( 0 );
    printf( "%d,%d→%d,%d¥n", bmp.Width, bmp.Height, x, y );
    Resize( bmp.Width, bmp.Height, x, y, fp, &bmp );
    fclose( fp );
    MFREE( buf1 );
}

unsigned short      d[2][513];
unsigned int        r[512], g[512], b[512];
unsigned int        rr[512], gg[512], bb[512];

void        Resize( int odx, int ody, int dx, int dy, FILE *fp, BMPHEADER *bmp )
{
    int     x, y;
    int     xx, yy, r0, g0, b0;
    int     sx01, sx02, sx11, sx12;
    int     sy01, sy02, sy11, sy12;
    unsigned short  *vp;
    unsigned char   *buf;
    BMPLINEBUF      lb;

    vp = (unsigned short *)0xC00000;
    vp += (dy-1)*512;
    lb.y = ody-1;
    GetBMPLineFirst( fp, bmp, buf1, &lb );
    Convert24bit( bmp );
    for( x=0; x<=dx; x++ ){
            d[0][x] = odx*x/dx;
            d[1][x] = odx*(dx-x)%dx;
            if( d[1][x]==0 ) d[0][x]--;
    }
    sy11 = -1;
    sy12 = 0;
    for( yy=0; yy<dy; yy++ ){
            sy01 = sy11;
            sy02 = sy12;
            sy11 = ody*(yy+1)/dy;
            sy12 = ody*(dy-yy-1)%dy;
            if( sy12==0 ) sy11--;
            for( xx=0; xx<dx; xx++ ){
                    g[xx] = gg[xx]*sy02;
                    r[xx] = rr[xx]*sy02;
                    b[xx] = bb[xx]*sy02;
            }
            for( y=sy01; y<sy11; y++ ){
                    sx11 = -1;
                    sx12 = 0;
                    buf = buf2;
                    for( xx=0; xx<dx; xx++ ){
                            sx01 = sx11;
                            sx02 = sx12;
                            sx11 = d[0][xx+1];
                            sx12 = d[1][xx+1];
                            gg[xx] = g0*sx02;
                            rr[xx] = r0*sx02;
                            bb[xx] = b0*sx02;
                            for( x=sx01; x<sx11; x++ ){
                                    b0 = *(buf++);
                                    g0 = *(buf++);
                                    r0 = *(buf++);
                                    gg[xx] += g0*dx;
                                    rr[xx] += r0*dx;
                                    bb[xx] += b0*dx;
                            }
                            gg[xx] -= g0*sx12;
                            rr[xx] -= r0*sx12;
                            bb[xx] -= b0*sx12;
                            g[xx] += gg[xx]*dy;
                            r[xx] += rr[xx]*dy;
                            b[xx] += bb[xx]*dy;
                    }
                    if( GetBMPLineNext( fp, buf1, &lb )==0 )
                            Convert24bit( bmp );
            }
            for( xx=0; xx<dx; xx++ ){
                    g[xx] -= gg[xx]*sy12;
                    r[xx] -= rr[xx]*sy12;
                    b[xx] -= bb[xx]*sy12;
                    g[xx] += odx*ody/2;
                    r[xx] += odx*ody/2;
                    b[xx] += odx*ody/2;
                    g[xx] /= odx*ody;
                    r[xx] /= odx*ody;
                    b[xx] /= odx*ody;
                    g[xx] += ((xx%2)*2+yy%2)*2;     /* dither */
                    r[xx] += ((xx%2)*2+yy%2)*2;
                    b[xx] += ((xx%2)*2+yy%2)*2;
                    g[xx] >>= 3;
                    r[xx] >>= 3;
                    b[xx] >>= 3;
                    if( g[xx]>31 ) g[xx] = 31;
                    if( r[xx]>31 ) r[xx] = 31;
                    if( b[xx]>31 ) b[xx] = 31;
                    *(vp++) = (g[xx]<<11)|(r[xx]<<6)|(b[xx]<<1);
            }
            vp -= 512+dx;
    }
}

void        Convert24bit( BMPHEADER *bmp )
{
    int     x, i;
    unsigned char   *bp1, *bp2;

  , bp1 = buf1;
    bp2 = buf2;
    switch( bmp->BitCount ){
    case 1: /* Monotone */
            for( x=0, i=0x80; x<bmp->Width; x++, i>>=1 ){
                    if( i==0 ){
                            i = 0x80;
                            bp1++;
                    }
                    *(bp2++) = *(bp2++) = *(bp2++) = (*bp1&i)?0xff:0;
            }
```

▶ペンギン情報コーナーを見て,「やった, 液晶のディスプレイテレビだ~」と思ったら, 実は液晶ディスプレイを使った大きめの液晶テレビだった。しかも, よく考えたらテレビのCMで宣伝していたやつだった。

森谷　好雄(18)北海道

```
		break;
	case 4: /* 16 Color */
		for( x=0; x<bmp->Width; x++ ){
			i = (x%1)?(*(bpl++)&0xff):(*bpl>>4);
			*(bp2++) = pal[i][0];
			*(bp2++) = pal[i][1];
			*(bp2++) = pal[i][2];
		}
		break;
	case 8: /* 256 Color */
		for( x=0; x<bmp->Width; x++ ){
			i = *(bpl++);
			*(bp2++) = pal[i][0];
			*(bp2++) = pal[i][1];
			*(bp2++) = pal[i][2];
		}
		break;
	case 24:/* Full Color */
		for( x=0; x<bmp->Width*3; x++ ){
			*(bp2++) = *(bpl++);
		}
		break;
	}
}
```

リスト10

```
#include	<stdlib.h>
#include	<stdio.h>
#include	<iocslib.h>
#include	<doslib.h>
#include	<string.h>
#include	"bmp.h"

void		WriteGLMHeader( int, int, FILE * );
void		ConvertGLM( FILE *, FILE *, BMPHEADER * );
void		Convert64k( unsigned char *, unsigned short *, BMPHEADER * );

unsigned char		BMPname[90], GLMname[90];
unsigned char		pal[256][3];
int		x1, y1, x2, y2; /* 対象領域 */
unsigned char		*buf1;  /* BMP生データ */
unsigned short		*buf2;  /* GLM変換データ .*/
int			size;   /* 確保できたメモリのライン数 */

void		main( int ac, char *av[] )
{
	FILE	*fp1, *fp2;
	BMPHEADER	bmp;
	int	i, x, flag1=0, flag2=0;

	printf( "BMP to GLM Converter v.1.00¥t(c)1995 Isawo-Kikuchi¥n" );
	for( i=1; i<ac; i++ ){
		if( av[i][0]=='/' || av[i][0]=='-' ){
			flag1 = 1;
			sscanf( &av[i][1], "%d,%d,%d,%d", &x1, &y1, &x2, &y2 );
		} else {
			switch( flag2++ ){
			case 0: strmfe( BMPname, av[i], "BMP" );
				break;
			case 1: strmfe( GLMname, av[i], "GLM" );
				break;
			}
		}
	}
	if( flag2==0 || flag2>2 ){
		printf( "[usage] BMP2GLM [/x1,y1,x2,y2] <BMPfile> [<GLMfile>]¥n"
			"¥tBMPファイルをGLMファイルに変換します。¥n"
		);
		return;
	}
	if( flag2==1 ) strmfe( GLMname, BMPname, "GLM" );
	fp1 = fopen( BMPname, "rb" );
	if( fp1==NULL ){
		printf( "BMPファイルがありません。¥n" );
		return;
	}
	if( GetBMPHeader( fp1, &bmp )<0 ){
		printf( "このBMPファイルには対応していません。¥n" );
		fclose( fp1 );
		return;
	}
	printf( "XSize:%4d YSize:%4d %2dBit¥n", bmp.Width, bmp.Height, bmp.BitCount );
	if( flag1 ){
		if( x1>x2 ){ i = x1; x1 = x2; x2 = i; }
		if( y1>y2 ){ i = y1; y1 = y2; y2 = i; }
		if( x1<0 || x2>=bmp.Width || y1<0 || y2>=bmp.Height ){
			printf( "指定領域範囲が不正です。¥n" );
			fclose( fp1 );
			return;
		}
	} else {
		x1 = y1 = 0;
		x2 = bmp.Width-1;
		y2 = bmp.Height-1;
	}
	x = (bmp.Width*bmp.BitCount+7)/8;
	size = MALLOC( 0x1000000 )&0xffffff;
	size -= x;
	size /= (x2-x1+1)*sizeof( unsigned short );
	if( size>(y2-y1+1) ) size = (y2-y1+1);
	if( size<=0 ){
		printf( "メモリが確保できません。¥n" );
		fclose( fp1 );
		return;
	}
	buf1 = (unsigned char *)MALLOC( x+size*((x2-x1+1)*sizeof( unsigned short)) );
	buf2 = (unsigned short *)&buf1[x];
	fp2 = fopen( GLMname, "wb" );
	if( fp2==NULL ){
		printf( "GLMファイルをオープンできません。¥n" );
		fclose( fp1 );
		return;
	}
	printf( "(%d,%d)-(%d,%d)→%dx%d GLM¥n", x1, y1, x2, y2, x2-x1+1, y2-y1+1 );
	WriteGLMHeader( x2-x1+1, y2-y1+1, fp2 );
	GetBMPPalet( fp1, &bmp, pal );
	ConvertGLM( fp1, fp2, &bmp );
	MFREE( buf1 );
	fclose( fp1 );
	fclose( fp2 );
}

void		WriteGLMHeader( int dx, int dy, FILE *fp )
{
	putl( 'GR65', fp );
	putl( dx*dy*2+8, fp );
	putl( 0, fp );
	putw( dx, fp );
	putw( dy, fp );
}

void		ConvertGLM( FILE *fp1, FILE *fp2, BMPHEADER *bmp )
{
	int	y, i, dx;
	BMPLINEBUF	lb;
	unsigned short	*bp2;

	dx = (bmp->Width*bmp->BitCount+7)/8;
	for( y=y1; y<=y2; y+=size ){
		if( size>y2-y+1 ) size = y2-y+1;
		lb.y = y+size-1;
		bp2 = buf2+(x2-x1+1)*(size-1);
		GetBMPLineFirst( fp1, bmp, buf1, &lb );
		Convert64k( buf1, bp2, bmp );
		bp2 -= x2-x1+1;
		for( i=1; i<size; i++, bp2-=x2-x1+1 ){
			GetBMPLineNext( fp1, buf1, &lb );
			Convert64k( buf1, bp2, bmp );
		}
		fwrite( buf2, sizeof( unsigned short ), (x2-x1+1)*size, fp2 );
		if( ferror( fp2 ) ){
			printf( "ディスクがいっぱいです。¥n" );
			break;
		}
	}
}

void		Convert64k( unsigned char *bp1, unsigned short *bp2, BMPHEADER *bmp )
{
	int	x, i;
	int	r, g, b;

	switch( bmp->BitCount ){
	case 1: /* Monotone */
		i = 0x80>>(x1%8);
		bp1 += x1/8;
		for( x=x1; x<=x2; x++, i>>=1 ){
			if( i==0 ){
				i = 0x80;
				bp1++;
			}
			*(bp2++) = (*bp1&i)?0xffff:0;
		}
		break;
	case 4: /* 16 Color */
		bp1 += x1/2;
		for( x=x1; x<=x2; x++ ){
			i = (x%1)?(*(bp1++)&0xff):(*bp1>>4);
			b = pal[i][0]>>3;
			g = pal[i][1]>>3;
			r = pal[i][2]>>3;
			*(bp2++) = ((((g<<5)|r)<<5)|b)<<1;
		}
		break;
	case 8: /* 256 Color */
		bp1 += x1;
		for( x=x1; x<=x2; x++ ){
			i = *(bp1++);
			b = pal[i][0]>>3;
			g = pal[i][1]>>3;
			r = pal[i][2]>>3;
			*(bp2++) = ((((g<<5)|r)<<5)|b)<<1;
		}
		break;
	case 24:/* Full Color */
		bp1 += x1*3;
		for( x=x1; x<=x2; x++ ){
			b = *(bp1++)>>3;
			g = *(bp1++)>>3;
			r = *(bp1++)>>3;
			*(bp2++) = ((((g<<5)|r)<<5)|b)<<1;
		}
		break;
	}
}
```

リスト11

```
#	BMP関連コマンドメイクファイル

LIBS = baslib.l iocslib.l doslib.l floatfnc.l
CFLAG = -c -Wall -O -fstrength-reduce -fomit-frame-pointer
AFLAG = /u /w
LFLAG = -z-heap=8192 -z-stack=4096

all : bmp256.x bmpfit.x bmp2glm.x

bmp256.x : bmp256.o PageSync.o bmpload.l
	gcc $(LFLAG) $^ $(LIBS)

bmpfit.x : bmpfit.o bmpload.l
	gcc $(LFLAG) $^ $(LIBS)

bmp2glm.x : bmp2glm.o bmpload.l
	gcc $(LFLAG) $^ $(LIBS)

bmpload.l : getheader.o getpalet.o getimage.o getlinefirst.o getlinenext.o
	lib $@ $^

%.o : %.c bmpload.h
	gcc $< $(CFLAG)

%.o : %.s
	has $(AFLAG) $<
```

PCMデータを拾い出す

データトラックの読み出し方

Taki Yasushi 瀧 康史

そのままではデータを読み込めないCD-ROMもある
そういったタイプのものに対応したツールを作ってみよう
AD PCMファイルを識別して自動で取り出すことも考えてみる

CD-ROMドライブを買おう

CD-ROMがデータ供給メディアとして王座に君臨したのは,「Windowsのソフトが CD-ROMで供給され始めてから」ではないかと個人的には思っています。公称値540Mバイトの容量とデュプリの安さから考えたらこの王座は黙っていても,誰がなんといっても,そうは簡単に崩れないかと思います。

Macintoshとのハイブリッドにすると半分ぐらいの容量になるため,ときどき容量的に足りないこともあるでしょうが,ここまで容量,コストを兼ねるものはそうはないでしょう。

CD-ROMソフトというのは「ゲーム」などの機種依存モノから,単なるデータ集である場合までいろいろです。パソコン用のCD-ROMを考えると,その多くはデータ集だと思います。雑誌の付録なども最近はCD-ROMが多くなってきています。私の購読している雑誌にもいくつかCD-ROMが詰まってきまして,格納ケースに困ることもしばしばです。自分のパーソナルマシンは,共にX68030(040)と,PC-9821Xa10なのですが,どちらもCD-ROMを装備しています。

そんななか,いまCD-ROMがついていないマシンは,X68000を含めても結構,使いづらいものがあります。特にその理由は,X68000にとってLANが縁遠い……という部分まで繰り下がるのですが,まあ,ここではおいておきましょう。

いいたいことは,要するにデータ環境を考えると,CD-ROMはどんなマシンでも必須かもしれないということですから。

最初に私がCD-ROMを買ったときには,かなりフロンティア的な思想がありましたが,いまではそうでもない時代になりました。すぐにはほしくなくても,CD-ROMはあったらいいな,みたいに考えている人もいますし,実際,友人と秋葉原を歩いていて,1万円ちょっとの格安CD-ROMドライブに食指が動いて衝動買いしてるのを,「しめしめ」と何度も見かけています(X68000のためのCD-ROMドライブです)。

これがあなたがCD-ROM導入を考えるきっかけになるかどうかはわかりませんが,今後,CD-ROMが面白いコンピュータライフを送るための必須アイテムであることは,間違いない事実なのかもしれません。

読めないCD-ROMたち

Windowsが普及したおかげで,読めないCD-ROMはずいぶん減ってきました。一時期はMacintosh用で妙なフォーマットがありましたが,いまではMacintosh用でもそれ相応のツールを利用すれば,出回っている大半のものを読むことができます。ただ,さまざまな理由から,MacintoshのCD-ROMは,やっぱりMacintosh以外で使うのは,向いてないとは思いますから,なんにせよ試すならWindows用のCD-ROMを購入したほうがいいでしょう。

ハイブリッドメディア/CD-ROMというのは,MacintoshでもWindowsでも読めるCD-ROMですから,X68000でも無難に読むことができます。

これとは別に,最近のゲーム機のCD-ROMはたいていはISO9660フォーマットなので,買ってきてすぐ覗くことができます。ときには,CD-ROMの中に,ゲームスタッフの「熱い思い入れ」が入っていることもありますよね。

要するに,だいたい世の中で出回っているCD-ROMはX68000で読めることになりますが,多量に出回っているわりに,X68000で読めないというものも存在します。それは,PCエンジンのCD-ROM²ディスクです。中古ゲームショップに行けば,ゴロゴロと転がっていますよね。音楽プレイヤーにかけると,1曲目には,「このCD-ROMはPCエンジンのCD-ROM² DISKです……」というアナウンスが入っています(ときどき,お遊びが入っていることもあります)。

ISO9660フォーマットのCD-ROMに場合,1曲目はコンピュータデータが入っていますが,このROM²の場合,1曲目はCD-DA,すなわち音楽トラックなのです。この音声を聞くとわかりますが,PCエンジンのCD-ROM²は2曲目にコンピュータプログラムが入っています。最近のCDプレイヤーならば,コンピュータプログラムが入っているトラックは勝手に読み飛ばすなり無音にするなりしているみたいですが,少しばかり古いCDプレイヤーだと見事に懐かしいあのピーガーノイズが鳴ります。場合によっては,これらはスピーカーを傷めてしまうことがあるので注意。きっと,1曲目にいきなりそういうコンピュータデータを入れると,素人が使うゲーム機には向かないのではないか? というのがNECの考えだったのでしょう。

最近のPC-FXでもこの考え方は残っているようで,基本的にはPCエンジンとPC-FXは変わらないみたいです。したがって,この2つのCD-ROMは,同じ手順で読み出すことができます。

データトラックの読み出し方

では,このデータトラックを読む方法ですが,意外にも簡単に読み出すことができます。libcであるなら,_scsi_read()命令で一発です。ただしCD-ROMドライブの構造上,CD-DA部(音声トラック)をデータとして読むことはできませんから,CD-ROM

固有の命令であるReadTOCなどを使い，2トラック目の最初の論理アドレスと最後の論理アドレスをCD-ROMから読み出さねばなりません。それには，前回紹介したCD-ROMドライブで，CD-DA部を再生するためのコマンドを利用します。

サンプルプログラムはdumppce.xといって，CD-ROM²やPC-FXのCD-ROMを読み出すものです。言語は，g++(Charlie版)を利用。ちなみにCharlie版はSX-WINDOW本に入っています。

分割ソースで，SCSIのコマンドの部分と，dumppce本体に分かれています。libscsiex.cというのは，私がSCSI関係のライブラリを個人的にまとめているもので，前回のものに加えて，今回はReadTOCの結果を論理アドレスにして返すルーチンなどを追加しています。

ReadTOCというCD-ROMへのコマンドを利用すると，CD-ROMの各トラックごとの構造がわかります。CD-ROM²は，2トラック目と決まっているので，2トラックの最初の論理アドレスを，ReadTOCで読み取ります。本来ならば，CD-ROMのサブコードなどを読めば，2トラック目がどこで終わるかわかるのかもしれませんが，今回は面倒なので，3トラック目の開始部分を終了ポイントにしています。

この値を利用して，_scsi_readコマンドでデータを読み出せば，容易にCD-ROM²からデータを吸い出すことができるのです。

PCMはいずこ?

サンプルプログラムを利用して，CD-ROM²からデータを吸い出すときには多少注意が必要です。冷静に考えればすぐわかると思いますが，CD-ROMに記録されているデータは強烈に大きなファイルを作成するからです。どうやら，CD-ROM²はミラーイメージとして，同じデータを複数保存しているっぽいですしね。

この膨大なデータのなかからPCMを探すのは難しいことです。ZVTが50Mバイトや60Mバイトのファイルを平気でアクセスできるならともかく，実際にはデータを少しずつ聞いて探さなければならないからです。

PCエンジンのPCMデータは，pcmplay.xで再生可能なので，サンプルプログラムで適当な大きさに切ってファイルに落とし，それをちまちま再生していきます。

とはいえ，たとえば150Mバイト分のファイルがあったら，そう簡単な話じゃないですよ。これはもう。何時間もピーピーガーガーいうノイズを聞き連ねても，なかなかPCMは出てきません。

そこで，プログラム側で当たりをつけることにします。PCエンジンのPCMは，X68000とほぼ同じなので，データにはX68000のものと同じような特性があります。一度でもサンプリングファイルのダンプを見たことがあればわかりますが，80や08という16進数データがたくさんあるのです。

AD PCMが4ビットです。つまりは，0と8が交互に出てくる可能性が非常に高いことになります。実は，この部分，AD PCMからリニアなPCMに直すと，無音部分を表しているのです。厳密にはX68000のAD PCMには無音部分はなく，多少のノイズが乗っていますが，結果的には0と8の羅列は無音を表します。

こういったところで使われるPCM波形というのは孤立的波形ですから，基本的には山があります。山以外は無音ですから，8と0が頻繁に出てきます。そこで，この8と0が頻繁に現れる部分を，抜き出してくれば，そこがPCMである可能性はかなり高くなってきます。

これが今回のプログラムに搭載された，簡易maddumpモードです。

これを緻密に行うのならば0と8の連続長を計る必要があります。最初の0と8の連続の部分で囲まれた部分を，サンプル的に16ビットPCMに変換し，その部分がある一定レベルをオーバーレンジするかどうかで，判断します。仮に0と8の羅列された部分に囲まれていても，オーバーレンジがそう起こる部分は，PCMである可能性は皆無に近く，したがって，PCMである部分として当たりをつけるには2つ条件を組み合わせると効果的です。

1)　8と0の連続データで囲まれている部分を探す(孤立波形かどうか?)

2)　囲まれた部分を一旦リニアのPCMに変換し，その部分がオーバーレンジするかどうか見てみる

以上でデータのなかからPCMを吸い出すことが可能になってくるのです。

プログラム利用法

ただし，上記の1)，2)を行うと，非常に時間がかかりますし，なによりも今回のように逐次処理をしている場合，不可能に近くなります。ですから，今回のサンプルでは1)のみを行ってみました。単純に頻度から割り出していきますが，頻度などは変えられるため，かなりPCM部分の当たりをつけることは可能です。

オプションは以下のものがあり，'-'以降の文字列で読み取ります。

●-IDn

CD-ROMのSCSI IDを指定します。nがID。これが設定されない場合，デフォルトでは6を指定します。環境変数CDROMが設定された場合その値を設定します。優先順位は　オプション-IDn＞環境変数CDROM＞デフォルトID:6の順です。

●-sizen

PCエンジン/FXのデータファイルは，ときに100Mバイトを超えるほど巨大になることがあるため，出力ファイルをn×2048バイト単位で分断します。指定された出力ファイル名の後ろに，数字が付加され区別されますが，Twentyoneなどを常駐しないと，区別できないおそれがあるので注意してください。

●-maddumpn

要するに自動解析モード。nは頻度の閾値。デフォルトは128。小さいほどデータの取りこぼしが少なくなる代わりに，余計なノイズが入りやすくなります。

●-mad-factor-n

発狂係数。デフォルトではn=4。データはプログラムの都合上，2048バイト単位で頻度を計ります。したがって，ちょうど山の中腹になってしまうデータもあるわけです。これを設定すると2048×nの分だけ，あたりとして発見された0,8の羅列のあとのブロックをすべてデータとして落とします。

●-start-address-sn

●-end-address-en

スタートアドレスsn，エンドアドレスen指定する。あたりをある程度つけてから，利用すると便利かもしれません。

●-help

ヘルプを出力する

●-verbose

おしゃべりモード。通常は指定しながら利用したほうがいいかもしれません。

Old Parrを飲みながら

どっかで予告していたとおり，結局98を買いました。ただ，予告どおりの「Pentium 75MHz+1280×1024ドット/フルカラーウィンドウアクセラ+64Mバイトメモリ」ではなくて，「Pentium100MHz(セカンドキ

ャッシュ256Kバイト）＋1280×1024ドット/フルカラーウィンドウアクセラ＋28800モデム＋1.2GバイトHDD増設＋16Mメモリ」になってしまいました。予定は変更されましたが，最初の予定どおり40万円ぐらいの出費です。

要するに，私が購入しようと思ったときにはPC-9821Xa7が売れ筋で手に入らなかったため，PC-9821Xa10を買ってしまったというわけです。HDDが多くなって，速度が上がった分，メモリが16Mバイトしかないためひぃひぃ状態です。なにやるにも……ってわけじゃないですが，PhotoCDの画面をいじくり回すプログラムを作ろうとすると，HDDをガリガリ……。

SX-WINDOWでのHDDスワップは憧れでしたが，ちょっとここまでスワップされてしまうと悲しくなっちゃいますね。本当ならば，メモリがドカン！　と増設できればいいのですけども。

X68030とXellent030(XVI専用)に，ツクモから増設16Mバイトメモリが出ますが，こちらへのプログラムの対応は簡単そうです。やっぱりデバイスドライバが揃ってるってのは強いかも。ソフトの対応はほとんど，数箇所書き直して，コンパイルするだけって感覚で，増設した16Mバイトメモリをアクセスできるようです。

ときに，シャープからの新しいマシンを求める声があるようですが，新しいマシンを求める声というのは，ある条件で２つに分けられます。それは「シャープのファン」であるのか，そうではなく「シャープに期待」している人なのか。

基本的に私は後者です。シャープの熱烈ファンというわけではなく，時代の流れの中でX68000を手に入れてみて，そのマシンアーキテクチャに惚れただけですから。だから，仮に次のマシンが出るにせよ，出なかったにせよ，それを買うか否かの判断は，「シャープだから」ではなく，「面白かったら」ということで考える予定です。

ただ，最近，私の心の中での問題ですが，パソコンを趣味として考えることを諦めてしまったようです。たとえばPC-9821Xa10を買って，いままでやりたくてもX68000ではできなかったことなど，いっぱいやっていますが，これは目的がコンピュータではなくて，目的は画像解析と音声処理です。

多分，私がMacintoshを日に日に嫌いになっていく理由も，そこに原因があるのだと思います。ですが，X68000はパソコンそのものが趣味でした。X68000を触ったから，シンセサイザに興味を持ったのかもしれませんし，X68000を触ったから画像解析に興味を持ったのかもしれません。

「趣味はパソコンで遊ぶことです……」

そういっていたあの時期が，非常に懐かしくなってしかたがないときがあります。過去を振り返るのは，個人的には好きなことじゃないのですが，思えばX68000の初期型が出たときもそんな感じでした。ただ，あの頃は，画像なり，音声なり，メモリなりほしいけどその当時のパソコンにはなかったことが「明確」でした。いまのこの感情の中では，それは不明確です。なにがほしいのか，なにを求めているのか。正直な話，わかっていないのです。

「シャープから新Xがでないかな……」

もし，あなたが単なるシャープファンでなくて，いっているのでしたら，あなたのこの声はどういう意味なのですか？

リスト1

```
/*
     PC-Engie & PC-FX CDROM Dump Program     for X68k

                                    copyleft by kohju.




$Id: dumppce.cc,v 1.1 1995/09/25 03:14:40 Kohju Exp $
$Log: dumppce.cc,v $
// Revision 1.1  1995/09/25  03:14:40  Kohju
// Initial revision
//

*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/dos.h>
#include <sys/iocs.h>
#include <sys/stat.h>
#include <sys/scsi.h>
#include <fcntl.h>
#include "scsiex.h"

struct    option{
     unsigned char     verbose;          // 0:無言実行  1:有言実行
     unsigned char     ID;               // CDROM SCSI-ID defaultは6。
     unsigned char     DataTrack         // データトラック番号、defaultは2。
     int                 size;             // size=0なら連続ファイル。
                                             // nならばn*2048単位で分断する。
     unsigned char     maddump;          // mad dump mode
     int                 thread;           // mad dump thread
     int                 madfactor;        // mad factor
     int                 startaddress;     // start address
     int                 endaddress;       // end address
};

     void    help( void );               // ヘルプ
     void    _main(option ,const char *);


int     main(int argc,char **argv)
{
     int       inc=0;          // インクリメント用変数
     char      *env_var;      // 環境変数 のためのポインタ
     option    opt={          // オプション列構造体
          0,                    // 無言実行
          6,                    // CDROMのSCSI-IDは6
          2,                    // DataTrack=2
          0,                    // split size
          0,                    // mad dump mode off.
          128,                 // mad dump thread.
          4,                    // mad factor=0
          0,
          0                     // startaddress & endaddress
     };

     printf(    "DumpPCE.x Dump read for PC-Engine CDROM2 and PC-FX CDROM¥n"
                "                                    copyleft by Kohju¥n¥n");

//                        環境変数

     if((env_var=getenv("CDROM")))          // CDROMがセットされてない場合無視
               opt.ID=atoi(env_var);          // セットされていたら、その値を利用

     if(argc==1){
          printf("Error:ファイル名を指定してください¥n¥n");
          help();
          exit(-1);
     }

//     オプション判別
     for(inc=1;inc<argc;inc++) {
          if(argv[inc][0]=='-'){        // オプションモード
               if(!strcmp(&argv[inc][1],"verbose")){
//                    printf("おしゃべりモード¥n");
                    opt.verbose=1;
               }
               else if(!strncmp(&argv[inc][1],"ID",2)){
                    if(argv[inc][3])     // 文字列がこれで終わりでなかったら……
                         opt.ID=atoi(&argv[inc][3]);
                    else                      // 終わりだったら次のinc++を読む
                         opt.ID=atoi(argv[++inc]);
                    //  上記の処理で、-IDの後に続く数字の前に、スペースがあって
も
                    //  なくてもよくなる。
               }
               else if(!strncmp(&argv[inc][1],"start-address-",14)){
                    if(argv[inc][15])    // 文字列がこれで終わりでなかったら……
                         opt.startaddress=atoi(&argv[inc][15]);
                    else{                     // 終わりだったらエラー
                         printf(     "start address値を指定してください。¥n");
                         exit(-1);
                    }
               }
               else if(!strncmp(&argv[inc][1],"end-address-",12)){
                    if(argv[inc][13])    // 文字列がこれで終わりでなかったら……
                         opt.startaddress=atoi(&argv[inc][13]);
                    else{                     // 終わりだったらエラー
                         printf(     "end address値を指定してください。¥n");
                         exit(-1);
                    }
               }
               else if(!strncmp(&argv[inc][1],"size",4)){
                    if(argv[inc][5])     // 文字列がこれで終わりでなかったら……
                         opt.size=atoi(&argv[inc][5]);
                    else                      // 終わりだったら次のinc++を読む
                         opt.size=atoi(argv[++inc]);
               }
               else if(!strncmp(&argv[inc][1],"maddump",7)){
                    if(argv[inc][8])     // 文字列がこれで終わりでなかったら……
                         opt.thread=atoi(&argv[inc][8]);
                    opt.maddump=1;
               }
               else if(!strncmp(&argv[inc][1],"mad-factor-",11)){
                    if(argv[inc][12])    // 文字列がこれで終わりでなかったら……
                         opt.madfactor=atoi(&argv[inc][12]);
                    else {                    // 終わりだったらエラー
                         printf(     "時間軸での救済率を指定してください¥n");
                         exit(-1);
                    }
               }
               else if(!strcmp(&argv[inc][1],"h")){
                    help();
                    exit(-1);
               }
               else if(!strcmp(&argv[inc][1],"?")){
                    help();
                    exit(-1);
               }
               else if(!strcmp(&argv[inc][1],"help")){
```

```
				help();
				exit(-1);
			}
		}
		else {	// ファイルの検索
			_main(opt,argv[inc]);
		}
	}
	printf("終了しました¥n");
	exit(1);
}

void _main(option opt,const char *ofile)
{
	_inquiry		scsiinq;
	unsigned long	cdrombaS;	// CDROMのblock address start
	unsigned long	cdrombaE;	// CDROMのblock address end
	int			mintrack;
	int			maxtrack;	// 最始トラックと最終トラック
	unsigned long	cdrombaSbk;

	int			outhandle;
	char			buff[2048];
	char			temp[256];
	int			fileinc=0;
	int			madflag=0;	// madflag.

	if(opt.verbose) {
//				012345678
		if(opt.maddump)
			printf(	"mad dump mode!¥n");

		if(opt.size) {
			sprintf(temp,"%s%03d",ofile,fileinc++);
			printf(	"outfile : %s¥n",temp);
			printf(	"split   : %d[bytes]¥n",opt.size*2048);
		}
		else{
			sprintf(temp,"%s",ofile);
			printf(	"outfile : %s¥n",temp);
		}
		printf(	"CDROM ID: %d¥n",opt.ID};
		if(opt.maddump){
			printf(	"mad fact: %d¥n",opt.madfactor);
			printf(	"mad thrd: %d¥n",opt.thread);
		}
	}

	while (_scsi_testunit(opt.ID) != 0) {
		printf("ID:%d の装置は操作不能です。¥n",opt.ID);
		exit(-1);
	}

	_scsi_inquiry(sizeof(struct _inquiry),opt.ID,&scsiinq);
	scsiinq.unit=scsiinq.unit & 0x1f;		// Qualifierを削除する
	if(scsiinq.unit !=5){
		printf("ID:%d の装置はCDROMではありません。¥n",opt.ID);
		exit(-1);
	}

	_scsi_ReadTOC_BA( opt.ID,0x00,&cdrombaS,&mintrack, &maxtrack );
						// 0x00は最始トラックを求めている
	_scsi_ReadTOC_BA( opt.ID,0xaa,&cdrombaE,&mintrack, &maxtrack );
						// 0xaaは最終トラックを求めている

	if(opt.verbose)
//				012345678
		printf(	"min trk : %d¥n"
				"DataAdrs: %#08x¥n"
				"max trk : %d¥n"
				"DataAdrs: %#08x¥n"
				,mintrack,(unsigned int)cdrombaS,maxtrack,(unsigned int)cdrombaE);

	_scsi_ReadTOC_BA( opt.ID, opt.DataTrack ,&cdrombaS,&mintrack, &maxtrack );
						// 0x02はPC-Engineのデータトラックを求めている
	_scsi_ReadTOC_BA( opt.ID,opt.DataTrack+1,&cdrombaE,&mintrack, &maxtrack );
						// 0xaaは最終トラックを求めている

	if(opt.startaddress)	cdrombaS=opt.startaddress;
	if(opt.endaddress)		cdrombaE=opt.endaddress;

	if(opt.verbose)
//				012345678
		printf( "DataTrk : %d¥n"
				"DataAdrs: %#08x～%#08x¥n"
				"DataSize: %#08x¥n"
				,opt.DataTrack,(unsigned int)cdrombaS,(unsigned int)cdrombaE,(unsigned int)(cdrombaE-cdrombaS));

	if((outhandle=open(temp,O_CREAT|O_RDWR|O_BINARY,S_IREAD|S_IWRITE))<0){
		printf("ファイルがオープン出来ません¥n");
		exit(-1);
	}

	cdrombaSbk=cdrombaS;
	while(cdrombaS<cdrombaE){
		if(opt.verbose)
//					012345678
			printf(	"ReadData: %#08x¥r",(unsigned int)cdrombaS);
		if(_scsi_read(cdrombaS++, 1, opt.ID, 3, buff)<0) {
			printf(	"Data Read Error¥n");
			exit(-1);
		}
		if(opt.maddump) {
			int		inc,freq=0;		// インクリメント用
			for(inc=0;inc<2048;inc++)
				if((buff[inc]==0x80) || (buff[inc]==0x08) ) freq++;
			if((freq>opt.thread) || madflag>0 ){
				if(madflag==0) {
					madflag=opt.madfactor;
					if(opt.verbose) printf("¥nSaveAdrs: %d¥n",(unsigned int)cdrombaS);
				}
				else if(opt.verbose) printf("saving..¥n");

				madflag--;
				if(write(outhandle,buff,0x800)<0){
					printf("ファイルが書き込めません¥n");
					exit(-1);
				}
			}
		}
		else if(write(outhandle,buff,0x800)<0){
			printf("ファイルが書き込めません¥n");
			exit(-1);
		}
		if(opt.size){
			if((cdrombaS-cdrombaSbk)%opt.size==0){
				close(outhandle);
				sprintf(temp,"%s%03d",ofile,fileinc++);
				if(opt.verbose)
//								012345678
					printf(	"¥nwrited  : %d[bytes]¥n"
							"out file: %s¥n"
							,opt.size*2048,temp);
				if((outhandle=open(temp,O_CREAT|O_RDWR|O_BINARY,S_IREAD|S_IWRITE))<0){
					printf("ファイルがオープン出来ません¥n");
					exit(-1);
				}
			}
		}
	}
	printf("¥n");
	close(outhandle);
}

void	help( void )
{
	printf(	"Help:¥n"
//			1	2	3	4	5	6	7	8
//		12345678901234567890123456798012345678901234567890123456789012345678901234567890
		"説明:このプログラムは、PC-EngineのCDROMイメージをそのままファイルに落とします。¥n"
		"  ファイル名は、出力ファイル名としてオプションで与えます。プログラムの性格上、¥n"
		"  出力ファイルは若干大きめのものになるので、注意してください¥n¥n"
		"書式:DumpPCE.x [オプション] 出力ファイル名¥n"
		"オプション:オプションは'-'以降の文字列で読み取ります。¥n"
		" -IDn		CDROMのSCSI IDを指定する。nがID。これが設定されない場合、デフォルト¥n"
		"		では6を指定します。環境変数CDROMが設定された場合その値を設定します。¥n"
		"		優先順位　オプション-IDn > 環境変数CDROM > デフォルトID:6¥n"
		" -sizen	PC-Engine/FXのデータファイルは、時に100MBを越えるほど巨大になること¥n"
		"		があるため、出力ファイルをn×2048バイト単位で分断する。¥n"
		"		出力ファイル名の後には数字が付加され区別されるが、Twentyoneなどを、¥n"
		"		設定しないと、区別出来ない恐れがあるので注意すること。¥n"
		" -maddumpn 発狂モード。nはthread。デフォルトは128。小さいほど救済率が上がる¥n"
		" -mad-factor-n¥n"
		"		発狂係数。デフォルトではn=4。時間軸での救済率¥n"
		" -start-address-sn¥n"
		" -end-address-en¥n"
		"		スタートアドレスsn、エンドアドレスen指定する。¥n"
		" -help		このヘルプを出力する¥n"
		" -verbose	おしゃべりモード¥n"
		);
}
```

リスト2

```
/*
  SCSIコール簡易操作マクロ Ver 0.5+
					copyleft by Kohju.
					参考 cd2pcm.x by WATA

	変なところが有ったらおしえてちょうだい……

$Id: libscsiex.cc,v 1.1 1995/09/25 03:14:48 Kohju Exp $
$Log: libscsiex.cc,v $
// Revision 1.1  1995/09/25  03:14:48  Kohju
// Initial revision
//

*/

#include <stdio.h>
#include <stdlib.h>
#include <sys/dos.h>
#include <sys/iocs.h>
#include <sys/scsi.h>

//								SCSIコマンド発行ライブラリ
int	scsi_cmdout_ex(
//	引数
		int		id,			// SCSI ID
		int		len,		// 送信データ長
	const void	*sendbuf,	// 送信データ配列
		int		rlen,		// 受信データ長
		void	*receivebuf	// 受信データ配列
)
{
	unsigned char status,msg;
	int req;

	if( (req=_scsi_select(id)) == 0) {			// アービトレーション&セレクションフェーズの発行
		if( (req=_scsi_cmdout( len, sendbuf )) == 0) {	// コマンドアウト
			if( rlen != 0 ) {
				while((req=_scsi_datain_p( rlen, receivebuf ))==0x8b){
					if((_iocs_bitsns(0x00) & 0x02) == 0x02) {
						printf("データインフェーズ発行中にESCキーが押されました¥n");
						exit(-1);
					}
				}
			if( req==0) {
				if( (req=_scsi_stsin( &status )) == 0 ) {		// ステータスインフェーズ
					if( (req=_scsi_msgin( &msg )) == 0) {		// メッセージインフェーズ
						req = (msg * 256) + status;
						return(req);
					}
					else {
						printf("メッセージインフェーズ中にエラーが発生しました¥n");
						return( -1 );
					}
				}
				else {
					printf("ステータスインフェーズ中にエラーが発生しました¥n");
					return( -2 );
				}
```

▶「沙羅曼蛇」の電源OFF時の声ですが，あれってエイリアンシンドロームの男のほうのキャラクター(名前忘れた)が，死ぬときに発する声と同じものではないでしょうか？
黒部　浩孝(22)三重県

```
                        }
                        else {
                            printf("データインフェーズ中にエラーが発生しました。エラ
ーコード=%d¥n",req );
                            if(req < 0 ) return(req);
                            if(( req = _scsi_stsin( &status ))==0){
                                if(( req = _scsi_msgin( &msg ))==0) {
                                    req = (msg * 256) + status;
                                    return(req);
                                }
                                else {
                                    printf("メッセージインフェーズ中にエラーが発生し
ました¥n");
                                    return( -1 );
                                }
                            }
                            else {
                                printf("ステータスインフェーズ中にエラーが発生しまし
た¥n");
                                return( -2 );
                            }
                        }
                    }
                    else {                  // 受信がないとき
                        if((req = _scsi_stsin( &status ))==0) {
                            if((req = _scsi_msgin( &msg ))==0) {
                                req = (msg * 256) + status;
                                return(req);
                            }
                            else {
                                printf("elserlen:メッセージインフェーズ中にエラーが発
生しました¥n");
                                return( -1 );
                            }
                        }
                        else {
                            printf("elserlen:ステータスインフェーズ中にエラーが発生し
ました¥n");
                            return( -2 );
                        }
                    }
                }
            else {
                printf("コマンドアウト中にエラーが発生しました¥n");
                return( -3 );
            }
        }
        else {
            printf("アービトレーション&セレクトフェーズ中にエラーが発生しました¥
n");
            return( -4 );
        }
};

void     _scsi_ReadTOC_MSF(     // ReadTOCを行いMSF形式で返す
        int                 ID,        // SCSI ID
        unsigned char    track,    // 開始トラック
        unsigned char    *M,        // M (戻り値としてのアドレス)
        unsigned char    *S,        // S (戻り値としてのアドレス)
        unsigned char    *F,        // F (戻り値としてのアドレス)
        int                 *min,     // 最始トラック (戻り値としてのアドレス)
        int                 *max)     // 最終トラック (戻り値としてのアドレス)
/*      説明:
        ReadTOC(43h)を実行しますが、戻り値を少なくするため、MSF形式に固定して
あります。
        また、手抜きで、トラックのコントロール属性を返していません。
    */

{
        unsigned char     sendbuf[10];     // 送信データ
        unsigned char     receivebuf[12];     // ローカルワーク

        sendbuf[0] = 0x43;           // scsi2 ReadTOC Command
        sendbuf[1] = 0x02;           // LUN=0,Set MSF Bit
        sendbuf[2] = 0x00;           // Reserved
        sendbuf[3] = 0x00;           // Reserved
        sendbuf[4] = 0x00;           // Reserved
        sendbuf[5] = 0x00;           // Reserved
        sendbuf[6] = track;          // 開始トラック
        sendbuf[7] = 0x00;
        sendbuf[8] = 0x0c;           // アロケーション長
                                         // イニシエータが受け取ることの出来るバイト長
        sendbuf[9] = 0x00;           // コントロールバイト
                                         // なにもしないので0。

        if( scsi_cmdout_ex( ID,10, sendbuf,12, receivebuf ) == 0 ) {
            *M   = receivebuf[9];          // M
            *S   = receivebuf[10];         // S
            *F   = receivebuf[11];         // F
            *min = receivebuf[2];          // Min. Track
            *max = receivebuf[3];          // Max. Track
        }
        else{
            printf(     "Read TOC(43h)がサポートされていません。¥n");
            exit(-1);
        }
};


void     _scsi_ReadTOC_BA(     // ReadTOCを行いBloack Addressで返す
        int                 ID,      // SCSI ID
        unsigned char    track,  // 開始トラック
        unsigned long    *ba,     // bloack address (戻り値としてのアドレス)
        int                 *min,  // 最始トラック (戻り値としてのアドレス)
        int                 *max)  // 最終トラック (戻り値としてのアドレス)
/*      説明:
        ReadTOC(43h)を実行し、ブロックアドレス、最始、最終を返します。
*/
{
        unsigned char     sendbuf[10];     // 送信データ
        unsigned char     receivebuf[12];     // ローカルワーク

        sendbuf[0] = 0x43;                 // scsi2 ReadTOC Command
        sendbuf[1] = 0x00;                 // LUN=0,unset MSF Bit
        sendbuf[2] = 0x00;                 // Reserved
        sendbuf[3] = 0x00;                 // Reserved
        sendbuf[4] = 0x00;                 // Reserved
        sendbuf[5] = 0x00;                 // Reserved
        sendbuf[6] = track;                // 開始トラック
        sendbuf[7] = 0x00;
        sendbuf[8] = 0x0c;                 // アロケーション長
                                               // イニシエータが受け取ることの出来るバイ
ト長
        sendbuf[9] = 0x00;                 // コントロールバイト
                                               // なにもしないので0。

        if( scsi_cmdout_ex( ID,10, sendbuf,12, receivebuf ) == 0 ) {
            *ba  = (receivebuf[ 8] << 24)
                   | (receivebuf[ 9] << 16)
                   | (receivebuf[10] <<  8)
                   | (receivebuf[11] <<  0);
            *min = receivebuf[2];          // Min. Track
            *max = receivebuf[3];          // Max. Track
        }
        else{
            printf(     "Read TOC(43h)がサポートされていません。¥n");
            exit(-1);
        }
};

int     _scsi_PlayAudioMSF(     // _scsi_PlayAudioMSFに従ってオーディオデータを再生す
る
//      引数
        int                 id,                    // CDROMのID
        unsigned char    SM,                  // スタートM (分)
        unsigned char    SS,                  // スタートS (秒)
        unsigned char    SF,                  // スタートF (1/75秒)
        unsigned char    EM,                  // エンドM (分)
        unsigned char    ES,                  // エンドS (秒)
        unsigned char    EF)                  // エンドF (1/75秒)
/*      _scsi_PlayAudioMSF(47h)を実行して、CDROMからオーディオデータを再生します。
        データはunsigned char形式で、分、秒、1/75秒で設定せねばなりません。    */

{
        unsigned char     sendbuf[256];

        sendbuf[0] = 0x47;                 //  Play Audio MSF Command
        sendbuf[1] = 0x00;                 // LUN=0,Reserved
        sendbuf[2] = 0x00;                 // Reserved
        sendbuf[3] = SM;                   // 開始:M
        sendbuf[4] = SS;                   // 開始:S
        sendbuf[5] = SF;                   // 開始:F
        sendbuf[6] = EM;                   // 終了:M
        sendbuf[7] = ES;                   // 終了:S
        sendbuf[8] = EF;                   // 終了:F
        sendbuf[9] = 0x00;
// コントロールバイト
        return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
};


int _scsi_PauseResume( int id ,unsigned char PR)
/*      説明
        CDROMをPauseしたりResumeしたりする。
        PR     0     ポーズ      べつにCDROMがナイスポージングを決めるわけではない
        PR     1     レジューム
*/

{
        unsigned char     sendbuf[256];
        sendbuf[0] = 0x4B;                 //  Play Audio MSF Command
        sendbuf[1] = 0x00;                 // LUN=0,Reserved
        sendbuf[2] = 0x00;                 // Reserved
        sendbuf[3] = 0x00;                 // Reserved
        sendbuf[4] = 0x00;                 // Reserved
        sendbuf[5] = 0x00;                 // Reserved
        sendbuf[6] = 0x00;                 // Reserved
        sendbuf[7] = 0x00;                 // Reserved
        sendbuf[8] = PR;                   // Pause & Resume
        sendbuf[9] = 0x00;                 // コントロールバイト
        return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
};


int _scsi_PlayAudioTrackIndex(
        int id,                             // ID
        unsigned char STrack,           // スタートトラック
        unsigned char SIndex,           // スタートインデックス
        unsigned char ETrack,           // エンドトラック
        unsigned char EIndex)           // エンドインデックス
/*
        スタートトラック、インデックスとエンドトラック、インデックスを指定して
、
        オーディオデータを再生する。
        再生は、PlayAudioTrackIndex(48h)をコールして実現している。
*/
{
        unsigned char     sendbuf[256];

        sendbuf[0] = 0x48;                 //  Play Audio MSF Command
        sendbuf[1] = 0x00;                 // LUN=0,Reserved
        sendbuf[2] = 0x00;                 // Reserved
        sendbuf[3] = 0x00;                 // Reserved
        sendbuf[4] = STrack;               // Reserved
        sendbuf[5] = SIndex;               // Reserved
        sendbuf[6] = 0x00;                 // Reserved
        sendbuf[7] = ETrack;               // Reserved
        sendbuf[8] = EIndex;               // Reserved
        sendbuf[9] = 0x00;                 // コントロールバイト
        return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
};
```

リスト3

```
/*
$Id: scsiex.h,v 1.1 1995/09/25 03:18:32 Kohju Exp $
$Log: scsiex.h,v $
 * Revision 1.1  1995/09/25  03:18:32  Kohju
 * Initial revision
 *
*/

        void     _scsi_cmdout_ex( int,int,const void *,int,void *);
        void     _scsi_ReadTOC_MSF(int, unsigned char,unsigned char *,unsigned char *, unsigne
d char *,int *,int *);
        void     _scsi_ReadTOC_BA(int,unsigned char ,unsigned long *,int *, int *);
        int         _scsi_PlayAudioMSF(int,unsigned char,unsigned char,unsigned char,unsigned
char,unsigned char,unsigned char);
        int         _scsi_PauseResume( int ,unsigned char );
        int         _scsi_PlayAudioTrackIndex(int,unsigned char,unsigned char,unsigned char,u
nsigned char);

```

リスト4

```
dumppce.x : dumppce.o libscsiex.o
        gcc2 dumppce.o libscsiex.o -ldos -liocs -lscsi

dumppce.o : dumppce.cc
        gcc2 -c dumppce.cc -O -Wall -m68000 -fomit-frame-pointer -fstrength-reduce -fforce-mem
-fforce-addr -finline-functions

libscsiex.o : libscsiex.cc
        gcc2 -c libscsiex.cc -O -Wall -m68000 -fomit-frame-pointer -fstrength-reduce -fforce-m
em -fforce-addr -finline-functions
```

▶10月号の20ページの上のほうは画面写真じゃないの？　プリンタ出力とは思えない。下のほうはエンボス地の紙にパステルとか水性色鉛筆で描いたような感じで，絵柄によってはこういうのもいいかも。
玉木　俊秀(27)神奈川県

CD-R導入の手引き

CD-ROMを制作する

Nakano Shuichi 中野 修一

「なければ作るX6800ユーザー」とはいうもののなかなか手が出ないCD-ROM
本格的なものは置いといて，まずは手始めとして
ライトワンスCDにX6800用のプログラムやデータを書き込んでみよう

編集部にCD-Rが装備された。

いきなりだが，CD-Rというのは，要するにCD-ROMを作成するための機械で，ライトワンスCD（その名のとおり，1回だけ書き込める）にデータを書き込むことができる。書き込んだものはそのまんまCD-ROMのように扱うことが可能だ。PCMデータとして用意してあれば音楽CDを作成することもできる。

従来は個人用途の製品とはいいがたいものがあったのだが，最近はかなり安価な製品が現れてきた。売値で15万円程度なら個人用途で使う人も出てくる値段だ。

しかし，残念ながらX68000に対応した製品というのは見あたらない（当然か）。ハード自体はSCSI接続なのでつなげばつながるのだが（CD-ROMドライブとして使用することはできる），書き込みには専用のソフトが必要だ。書き込みのコマンドがわかればSCSIコールで直接送ってやるということもできるのだろうが，最近の機器のマニュアルはそういったのが載っているほど親切ではない。やむなし。

ということで，ここにたまたまCD-Rを使用するためにAT互換機ごと買ってしまった御仁がいたというわけだ。CD-Rドライブが秋葉原価格で139,800円，AT互換機が250,000円程度……（Pentium-120MHz，EDO16Mバイト，HD1.4G）。まあ，好き好きというものであろう。

X68000用としては究極の周辺機器ともいえるCD-R。ここではもの好きな人のためのCD-R導入の心得をまとめてみよう。本来X68000とはあまり関係のない話なので，あくまでも参考までに。

ハードウェア環境

まず，CD-RドライブとX68000以外のマシンが必要だ。今回使用したのはキャラベルデータシステムのCDR-2Xというドライブだ。この製品は，PC-9801，Macintosh，Windowsマシンと，AMIGAやX68000以外のほぼすべての機種に対応している。

AT互換機が安いとはいうものの，まわりになにかのパソコンがあったらそれを利用することを考えたほうが利口だろう。必要なのは，書き込みソフトが動く環境とSCSIカード，CD-ROMの倍くらいの容量のハードディスク，データ運搬用のMOなどといったところだ。

今回はAT互換機に接続して使用しているので，Windows上での操作を中心に解説していくことにする（うーむ）。

まずマシンだが，SCSIの転送速度さえ速ければ，特にCPUパワーは必要ではない。CD-Rは書き込み時に一定の転送レートでデータを要求してくる。ここでデータの送り側が追いつかなければ，書き込みミスが発生して，そのメディア1枚が丸ごとオシャカになる。

こういった事情があるため，書き込み時にはCD-ROMに書き込む形式のデータを先に作っておいて，それを書き込んでいってやるわけだ。CDに書き込む形式のデータというのを少し説明しておこう。CDというのはエラーが発生することをあらかじめ考慮されたデータフォーマットで記録されている。補正用のデータ（ECC）を含んだ状態で記録されているわけだ。通常の音楽CDもそうだし，CD-ROMのデータ部分はエラー訂正符号が音楽用よりさらに強化されたものが使用されている。

最近のCD-R用ソフトでは，ディスク上のディレクトリ構造をそのまま指定して，リアルタイムにそういったデータ形式に変換しつつCD-ROMに書き込むというモードもついていることが多いのだが，これは各種機器が十分に高速な状態でないと機能しない。先ほど挙げたマシンで試してみたところ，途中で書き込みエラーが発生した。CPUパワーには問題がないと思われるので，変換時のデータのあるドライブとワーク用のドライブが同じだったのがまずかったのだろう。

いずれにせよ，結構デリケートな問題なので，安全を期するならいったん全データを変換してから書き込んだほうがいいだろう（その分ハードディスク容量は食う）。逆にいえばハードディスクさえ十分に用意すれば，それほど高速なマシンは必要ないということである（変換時間はかかるが）。

書き込みツールはWindows上で動作する



CDR-2X　キャラベルデータシステム

ので，Windowsが動けば特に機種は問わない(はず)。しかし，Windows上のツールとはいうものの，なぜかコンベンショナルメモリ（MS-DOSでのフリーエリア）が600Kバイト程度必要ということなので，MS-DOSの時点でフリーエリアをできるだけ稼いどかなければならない。非常にダサいけど，この世界ではこんなのが結構まかり通ってるんだよなあ。

このあたりは購入するドライブによって添付ソフトが違うと思われるので，ほかの製品では状況が変わってくるだろう。CDR-2Xに付属のものはB'sRECORDER，ビー・エイチ・エー社の製品だ。使い方などはマニュアルを見て操作するだけだが，はっきりいってわかりにくいし，無駄も多い。まあ，Windowsだからしかたないか。

書き込みの注意

さて，このツールを使うためにはX68000からAT互換機へファイル転送しなければならないわけだが，まず，気をつけなければならないのがファイル名の問題だ。MS-DOSでは８文字＋３文字でファイル名がぶった切られるので，あらかじめ適当なものに直しておくほうがよい。当然，小文字だとMS-DOSでは読めない。COPYALLの/Uオプションでファイル名を大文字化しながらMOなどへ転送するのがよいだろう。大文字しか読めないのにWindowsでは小文字表示になるのはなにか解せないものがある。

それはともかく，そもそもISOのCD-ROM規格では漢字も使用できないので，漢字を使ったファイル名も改めておこう。無理にISO規格にあわせる必要もないが，こういうものはできるだけ汎用的に作るのが筋である。

なお，計測技研のドライバではマルチセッションに対応していないため，CD-R用として使うにはいまひとつおいしさに欠けるところがある。マルチセッションというのは，要するに追記のことで，一度途中まで書き込んだCD-Rの後ろに新しいデータを書き加えていくことを指す。これにはCD-ROMドライブ，ドライバ双方が対応している必要がある。

一度にそんなに大量のデータを揃えるのは大変なので，CD-ROMを制作するときにはちびちび追記できたほうがありがたい。追記に対応できればハードディスクのバックアップ用などにも使用できるだろうに。

ちなみに，編集室のCDR-2Xで何度か追記を試みたがいずれも失敗に終わっている



EX-Systemだとこの程度

(できるはずなんだが？)。

使っていてもうひとつ問題があった。

おそらくはWindowsのせいだと思われるが，ファイル選択時に自動的にソートされてしまう。通常は書き込みたいディレクトリを指定するのだが，その中にあるファイルを無条件にアルファベット順に並べてくれるお利口さんな仕様のため，思ったようなディレクトリ構造が作れない。

そういえば，ひと月ほど前，最近Windowsを使い始めた伊滋見氏がファイルマネージメントのあるべき姿について滔々と講義してくれたのは，こういったことをいっていたのだとようやく気づいた。

製造コスト

ちなみに，CD-Rの生メディアは１枚約1,000円程度（秋葉原価格）だ。メディアには60分(540Mバイト)と74分(650Mバイト)の２種類があるが値段は特に変わらないようだ。メディアの信頼性や対応ドライブの多さなどを考えると，よほどのことがない限り74分を使う必要はないだろう。

最近128MバイトのMOは700円程度(秋葉原価格）になっている。たとえば，巨大同人ソフト……クラスのものではMOを使用したほうが有利かもしれない。X68000ユーザーではMOの所有率が非常に高いのも事実だから。あとは複製の手間だけである。

１枚ずつCD-Rで焼くのではなく，業者に頼んでプレスすることでCD-ROMの単価は大幅に下げることもできなくはない。ただし，この場合はプレス枚数が問題になってくる。CD-ROMのプレスには，まずプレスする型となる原版を作る必要がある。これがマスタリングという工程で，だいたい130,000〜150,000円くらいかかる。これさえできてしまえば，あとのプレスは１枚当たり180〜250円くらい(レーベル印刷，パッケージングを含む。なお，プレス枚数によってかなり変わる）となっている。計算してみるとCD-Rメディアの単価と比べて割安になるのはだいたい150〜200枚以上の場合に限られる（それでも１枚１枚焼く手間を考えれば安いか？）。

ま，同人ソフトにしてもCD-ROM化することは不可能ではないが，まだ少し現実的ではないかな？　といったところだ。プレス業者の広告を見ているとだいたい100枚単位で扱うのが相場のもののようだ。

個人ベースでCD-ROMソフトを作成する場合の目安にはなるだろうか？

その可能性

X68000の場合，Human68kを付属させた製品販売が認められている。これによりソフトを買ってきてすぐ起動できるなどのメリットがたくさん存在する。うまくすれば，CD-ROMからの起動もできるかな，と試してはみたのだが，残念ながらCD-ROMから起動することはできないようだ。もともとCD-ROMというものが単純なSCSIだけでなく，特殊なドライバを必要としていることから考えても無理っぽくはあったのだが。

ちょっと大がかりになるが，音楽CDの編集ということもできそうだ。CD2PCM.Xでデータを取り出してWAVファイルに変換してやれば，CD-Rで書き込むことができるので，気にいった曲を集めて１枚のCDにするといった使い方もできるだろう。ちと危ない使い方だが，音楽CDの場合デジタルコピーが第１世代まで広く認知されているので，まあ問題はないだろう。

そのほかでは，PhotoCDは大きなデータの圧縮方法がいまいちわからないため断念。PCDデータだけを作成するならともかく，ほかにもスクリプトがあるみたいだしなあ。パソコンに限らず汎用のグラフィックデータ供給媒体としてはいちばんよさそうなんだが（基本データサイズが768×512ドットだし）。

＊　＊　＊

試しに手持ちのグラフィックデータをMO３枚分とEX-System暫定版も込みで焼いてみた。しかし，まだ400Mバイトに満たない。

音楽データやグラフィックデータはいずれちゃんと整理したのを焼いてみようとは思うが……（あちこちから別々にデータが持ち込まれるので整理が大変)，まあ少し暇になってからだな。

CD-ROMを作るってのは考えればそれなりに可能性がある試みなのだが，独立したメディアとしては時期尚早，とりあえずあふれかえるデータをまとめるってところだけでも十分に意義がありそうだ。

試用レポート

Xellent30PRO

Kikuchi Isao 菊地　功

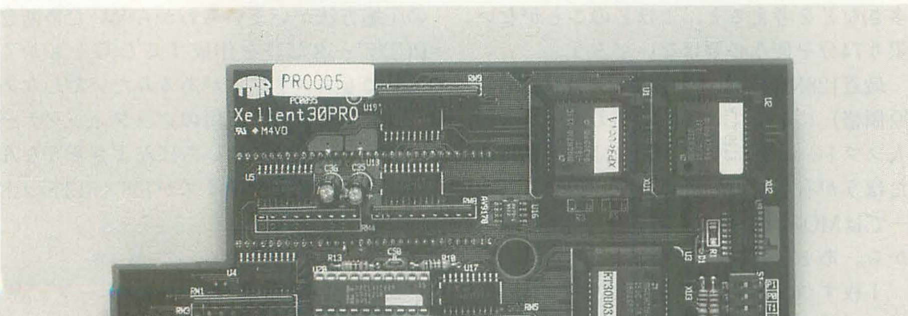
いままで互換機だなんだと迫害されてきたX68000 PRO/PROⅡ(以下PRO/PROⅡ)ユーザーに朗報です。前々から噂のあった，PRO/PROⅡ用のXellent30PROがついに発売になりました。私も，以前の2種類のXellentをレポートした紀尾井誠氏や瀧康史氏同様，X68030ユーザーなので(クロックアップはしてませんが)，はっきりいってそれほど有効とは思えないのですが，Xellent30(s)がそこそこ売れているようですので，きっと10MHzユーザーから見ればそれなりに満足できる代物なのでしょう。まあしかし，手軽に68030気分を味わいたいというのであれば，正しい選択なのかもしれません。SRAMを有効に使えば，それなりのパフォーマンスも期待できますしね。

基本スペック

基本的なハードウェアスペックは，先に発売されているXellent30sとまったく同じです。ドーターボード上には，システムクロックの2倍の20MHzで動作するMC68EC030-25と，33MHz非同期で動作するMC68882，それにローカルSRAMが搭載されています。ひょっとして，SRAMの容量が増えてないかとか，増設できるようになってないかなどの期待がもたれますが，残念ながら前回同様256Kバイト増設不可です。

Xellent30(s)同様，Human68k ver.3.0が必要で，SRAMを使ったXellent30(s)用のソフトウェアも問題なく動くはずです。ドライバ類もXellent30(s)のものが使える……はずなのですが，編集部に届いたものは，試用版ということもあるのかもしれませんが，標準の起動MPU選択メニュープログラムch30.sysでは，起動しませんでした。改良版であるch30_kai.sysを使えば起動はできたのですが，やはり微妙なタイミングの関係なのでしょうか。

PRO/PROⅡは，ほかの機種に比べて，クロックアップ改造をしている比率が少ないとは思うのですが，そういったマシンにXellent30PROを搭載するには，それなりの覚悟がいります(注：もちろん改造したものについて同社では動作保証していません)。ただ，PRO/PROⅡはもともとほかの機種に比べて個体差が激しく，Xellent30PROもある程度余裕をもたせて作ってあるようなので，MPUを速いものに載せ換えてやれば，結構動いてしまうかもしれません。責任はもてませんが。



取りつけ

取りつけにはカバーを外さなければなりませんが，PRO/PROⅡはほかの機種に比べて非常に開けやすいので，特に問題はないでしょう。ねじ数本で簡単に開いてしまうなんて，某国民機と錯覚してしまうほどです。MPUはスロットの手前の広い空間の中央あたりに位置していますので，まず筐体上部にねじで固定されている2本のフレームをはずします。1本だけでもスペース的に不可能ではありませんが，ねじ数本をはずす手間を怠って事故を起こすのも馬鹿らしいので，両方外すことをお勧めします。メモリを増設している場合はメモリも，ハードディスク内蔵モデルではハードディスクも外すことになるでしょう。MPUを慎重に抜き，抜いたMPUをXellent30PRO上の空いているソケットに向きに注意してしっかりと差し込みます。次にXellent30PROを本体に取りつけるのですが，このとき，メイン基板を真横から見ることができないので，ピンを間違えないように注意してください。歯医者さんが使うような鏡でもあれば少しは楽かもしれません。と思ったら，マニュアルにはフロントパネルも外すように書いてありました。なんだ，簡単に外れるんだ。そうすれば，基板を横から見られるし，事故を防げそうですね。そうそう，Xellent30PROには，メイン基板上に載っているコンデンサがぶつからないように，2カ所に穴が開いているのですが，ちゃんとその穴にはまるように，コンデンサが傾いているときには慎重にまっすぐにしておきましょう。このあたりは結構勇気のいる作業です。

取りつけが終了したら，コンセントを差し，フロントの電源を入れてすぐに切ってみましょう。緑のPOWERランプが点滅してから電源が落ちれば，取りつけは成功です。そうならない場合は，正常に動作していませんので，もう一度差し込みを確認してください。取りつけに成功しても，とりあえずカバーは外したままにして，モニターなどをつなぎ，起動させてみます。チェックプログラムを実行して異常がなければ，

起動メニューをSRAMに登録し(SRAM揮発癖のあるマシンはなんらかの対処が必要です)，どちらのMPUからでも起動が確認できれば問題ないのですが，個体差によってはデフォルトでは立ち上がらない場合があるようです。PRO/PROⅡは特に個体差が激しいので，Xellent30PROでは，内部タイミングを設定するディップスイッチが用意されています。これを適当に変えてやるのですが，マニュアルにはタイミングA，B，Cと書いてあるだけで，なんのタイミングかは記されていませんでした。また，ディップスイッチを変えて動かない場合でも，編集部の例のように，ch30.sysをch30_kai.sysなどに変えれば動くこともあるかもしれません。それでも動かない場合は……同社に連絡してみましょう。

パフォーマンス

とりあえず，stanfordベンチマークの結果を，ほかの機種と比較してみました(MPUが68000の場合はfloat2.xを，68030の場合はfloat4.xを使用)。

		int	float
X68000 PRO	10MHz	571.5	2487.7
Xellent30PRO	20MHz	283.9	1695.5
Xellent30s	20MHz	267.1	1526.8
X68000 XVI	16MHz	334.9	1473.8
Xellent30	33MHz	157.5	964.0
X68030	25MHz	107.3	673.5

値そのものにはあまり意味はありませんので，割合で比較してください(値が小さいほうが速い)。floatがもう少し速くなってもいいような気がするのですが，このベンチマーク自身がfloatの測定に分岐などのfloat以外の命令を多用しているということがあるのでしょう。整数演算を比べれば，10MHzの場合の約2倍の値が出ています。また，XVIよりも速いという結果も出ていますが，実際のアプリケーションではメモリウェイトなどの関係で，体感速度では10MHzの1.5倍程度，XVI未満といったところでしょう。

ちょっと気になるのが，Xellent30sよりも若干遅いということです。何度もいうようですが，PRO/PROⅡは個体差が激しいため，余裕をもたせて作ったからだそうです。特にPROⅡで「はまった」そうなのですが，PROと当たりのPROⅡでは，細工すればもう少し速くなるという話もあります。もちろん，ここから先の部分は各自の責任において試してください。

実際に使うには，Xellent30sとほぼ同じと思っていいでしょう。試しに「SUPER STREET FIGHTER Ⅱ」をプレイしようと思ったのですが，編集部のPROはメモリが2Mバイトしかなかったので，「STREET FIGHTER Ⅱ'」を起動させてみました。しかし，なぜかそれも起動しませんでした。どうやらCPU判別で030であることを識別し，多重PCMドライバを組み込むまではいいのですが，そのドライバがかなりメモリを消費し，メモリ不足に陥っているようでした。現に1声だけのPCMドライバを組み込むようにすれば，ゲームはできるようになりましたが，せっかく030になったのに，これではちょっと悲しいものがあります。ということで，Xellentを載せるなら(他機種でも)，メモリ増設は必須ですね。

肝心のゲームのほうは，もともとがそんなに重くなかったので，よくわかりませんでした。Xellent30sで「SUPER STREET FIGHTER Ⅱ」が目に見えて速くなったのだから，同じくらい速くなるはずですが(レポートになってない)。

まとめ

買いか？　と問われると，ちょっと悩んでしまいます。Xellent30sのときには，私としては「まあ，買いかな」と思っていたのですが，PRO/PROⅡとなると，「Mach-2」も非対応ですし，時期的な問題もあります。しかし，そういった背景がありながら，あえて「PRO/PROⅡ専用版」を出したメーカー側の勇気も賞賛に値するものがあります(しかも価格はそのまま)。Xellent30sを買ったユーザーは結構満足しているようですし，買っておいて損はないかもしれませんね。

Xellent30PRO　54,800円(税別)
東京システムリサーチ　☎0425(28)1824

ツクモの16Mバイトメモリボード

メモリが足りない！　という切実な声に反応してくれたのか，ツクモさんから，X68030とXellent30の2機種専用に，増設メモリボードが出ます。締め切り直前で入手したため，12月号で詳しくレポートしますが，簡単な概要を話してみましょう。

残念ながら，Xellent30sやXellent30PROには対応しておりません。040turbo利用時も，物理的な面で支障が出てしまいます。

対応機種は上の2機種ということで，わかる人にはわかりますよね？　CPUがMC68030用ということです。つまり，従来の16Mバイトのその上のアドレスにメモリがマッピングされています。こういうのをハイメモリといったりします。つまり，MC68000では存在しないアドレスに接続されているので，CPUがMC68030であることが前提条件になるのですね。アウトポートを叩いて云々というような，バンクメモリではないからです。ただ，拡張スロットには，16Mバイト以上のメモリをアクセスするアドレスが出力されていませんから，CPUの接続の合間にボードを差し込んで利用します。CPUからしか見えないところにハイメモリが16Mバイト存在するため，DMAのアクセスはできません。したがって，直接AD PCMなどのバッファにすることはできません。

ハイメモリはプログラムを作らないとアクセスできませんが，このメモリボードには，HIMEM.SYS(笑)という名前のメモリマネージャと，それに対応するRAMディスクドライバしか添付されていないようです。ただし，ドライバを利用するのは簡単ですし，C言語プログラマやアセンブラプログラマからは，利用しやすいように，作られているため，プログラムの対応は非常に簡単なはずです。C言語だったら具体的には，頭に1行追加し，命令を一部書き換えるだけで対応プログラムが作れてしまうぐらいです。EX-Systemもきっと対応してくれるんじゃないかな(未確認)。

メモリを有効に活かすためには，対応ソフトを利用するか，もしくはいままでメインメモリに常駐していたものを外に逃がすことを考えねばならないでしょう。具体的には，RAMディスク，ディスクキャッシュ，大きめのデバイスドライバなど，いくつか常駐してほしいものがありますが，対応は割と簡単だと思います。

詳しいことはまた来月。　(瀧　康史)



MODELのすすめ

プロジェクトチームDōGA
かまた ゆたか

CGAシステムのモデリングツールとして，CADが長年君臨してきましたが，それに代わる新しいモデラ，その名もMODELがいよいよ本格的に登場します。今回はCADとの違いを中心に，その実力を解説いたします。

はじめに

前回はCGA作品におけるテーマなどについて，偉そうなことを書きましたが，それについていろんな方から手紙や電話をいただきました。また，ネットなどでも討論されていました。

皆さんから，あの点については大いに賛同するがこの点については別の意見があるとか，自分は自分なりの道を進むとか，いま制作している作品はまさにあの問題を解決する作品だとかいった話を聞くことができました。

私としては，賛同するしないは関係なく，作品の内容を高めるためにはどうしたらよいか，考え，議論していただけて，たいへんよかったと思います。

さて，久しぶりなのでその名を忘れてしまった方もいると思いますが，近々，CGAマガジンの最新号が発行されます。もしかすると，本誌が出ているころには，もうTAKERUで発売されているかもしれません。

詳しい経緯は，コラムのほうをご覧いただくとして，このCGAマガジンには，ここ1年でバージョンアップされたツールをまとめて収録してあります。そこで，本誌においてもそれらのツールを取り上げたいと思います。

MODELのメリット

今回収録されるツールがどのようなものになるか，これを書いている時点では完全に決まっていませんが，目玉となるのは「MODEL.X」でしょう。このMODELは以前から発表されていましたが，まだまだ未完成で，実際に使っている人はいないと思います。

しかしその後，何度もバージョンアップされ，かなり実用性がでてきました。CADと比較して，MODELが圧倒的に優れているのは，まさにバージョンアップができるという点です(CADはソースが紛失している)。ですから，現時点で同等であっても，今後のことを考えると，そろそろ乗り換えてもよいでしょう。現にプロジェクトルームにおいては，すでにCADよりもMODELのほうが主流になっています。

MODELとCADを比較すると，機能的な面でMODELのほうが優れています。CADの機能のほとんどすべて(誰も使わないような機能を除く)がMODELで実現されているうえ，MODEL独自の機能も豊富にあります。特に，編集機能はだいぶ強化されました。

もうひとつ大きなメリットとして，精度が高くなりました。CADだと，たとえば正多角形を作画する場合，Z＝0の面で作成したはずなのにZ座標が1の頂点になったり，前後左右が微妙に非対称になってしまいます。そういった誤差がなくなりました。

それに対してデメリットもちょっとあります。まず，速度が遅くなりました。精度を上げたこと，移植性を上げたことなどが原因です。しかし，CADをX68030で動かすと，むしろ速すぎて困ることがありますので，MODELでちょうどいいぐらいです。ただ，初代機ではちょっとストレスが溜まるかもしれません。

そして，これはデメリットではないかもしれませんが，マウス操作がCADと異なるため，戸惑ってしまいます。これがいままでMODELが使われなかった最大の理由でしょう。しかし，これは慣れの問題であり，最初はつらくても，1週間も使えば気にならなくなります。

起動

CADと同じように，

MODEL ↵

MODEL 形状ファイル ↵

で起動できるほか，

MODEL アトリビュートファイル ↵ ……①

MODEL 形状ファイル アトリビュートファイル ↵ ……②

という形式も有効です。

共通のアトリビュートを使う物体を複数作成する場合，①の形式で起動すると，いちいちアトリビュート名を登録しなくてすみます。

単にアトリビュート名の情報を知りたいだけなら，②の形式は不要です。実は，MODELでは，アトリビュートで指定された色でそのポリゴンを描きます。つまり，赤

いポリゴンは，赤いラインで表示されるわけです。もっとも，8色しかありませんので，いちばん近い色となります。写真1が，②の形式で，起動した状態です。

起動はCADより時間がかかります。これは，MODELがマクロ言語をもっており，その言語で記述されたいろんな機能を読み込むからです。つまり，あなた自身，このマクロ言語を記述すれば，コンパイルすることなしに，自分の手でバージョンアップすることができます。詳しくはまた今度。

それからMODELでは，ひとつのファイルに複数の形状があっても，ちゃんと処理することができます。たとえば写真1の右側の下のウインドウに「body」と「wing」と「canon」が表示されていますが，これは，ロードされたファイルに3つの形状が存在していたことを意味します。この状態では，3つの形状が同時に表示されていますが，この中の特定のオブジェクトだけを表示したり，編集することももちろんできます。

初期設定

MODELを起動してその画面を見ると，CADと比べて，なにかもの足りない気がするでしょう。そう，3面図にグリッドが表示されていないのです。

また，CADの場合，デフォルトでグリッドが40，マウスによる十字カーソル移動量が10となっています。このあたりの設定に慣れている人には，取っつきにくく感じるかもしれません。ということで，まずはこのあたりの設定を，CADと同じようにしてみましょう。

「Window」のメニューの「Size...」を選択します。すると，図1のようなウインドウが開きます。

まず，「ズーム」ですが，これはCADの「scale」に相当し，3面図の表示範囲を設定する値です。この値が「4」ということは，表示範囲が600×400程度となり，CADの初期画面とだいたい同じです。

「グリッド」が，グリッドの表示とその細かさを設定する値で，1では表示しませんので，つまみを動かしてやります。CADのデフォルトは「40」ですが，MODELでは，20～50の間の値はとれません。ここは，「50」ぐらいにすればよいでしょう。

最後に「マウス」で十字カーソルの移動量を設定します。これはCADと同じく「10」にできます。ただ，CADの場合，マウスで「10」単位で動かし，キーボードの操作に「2」単位で動かすということができましたが，MODELには，(まだ)キーボードで動かす機能がありませんので，ちょっと小さめの「5」にしておくのもよいかもしれません。

しかし，十字カーソルの移動量の設定が，「WINDOW」の「ウインドウサイズの設定」の「マウス」ってのは変ですね。このあたりは，今後改善していかなければいけないでしょう。

アトリビュート設定

CADを起動して，最初にすることといったら，普通，アトリビュートの登録でしょう。MODELでは，「起動」のところで説明したとおり，アトリビュートを読み込んで起動する方法もありますが，いちから作成する場合は，やはり，アトリビュート名を登録しないといけません。「File」の「New Attr...」で図2のようなウインドウが開きます。

カラーコードというのは，表示されるラインの色です。1が青で，7が白になります。

このようにして登録されたアトリビュートは，画面の右側に一覧表示されます。ですから，表示されたこのアトリビュートのどれかをクリックし，反転させた状態で

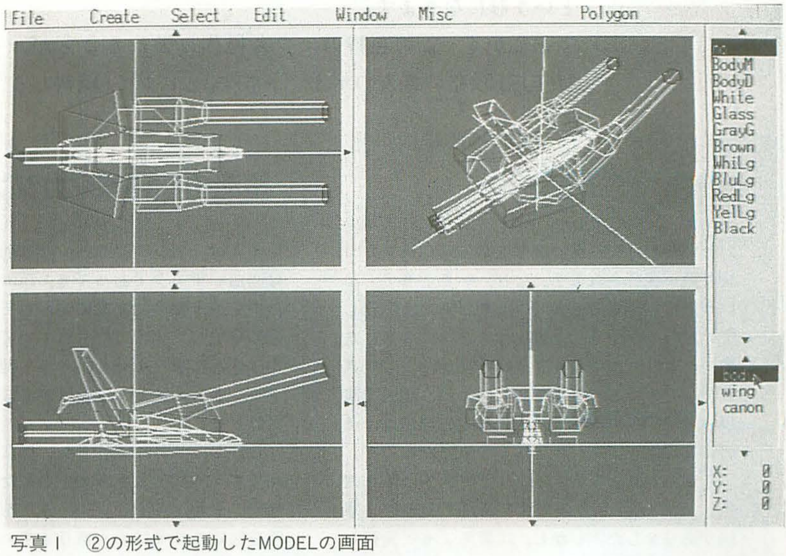


写真1　②の形式で起動したMODELの画面

図1　サイズ設定ウインドウ

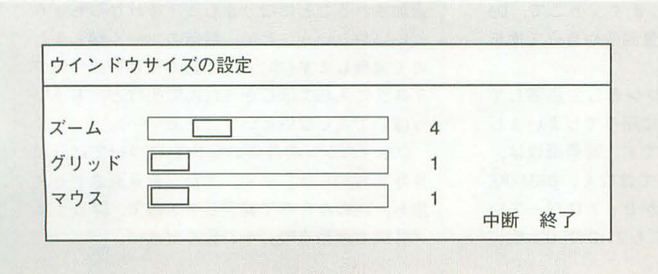
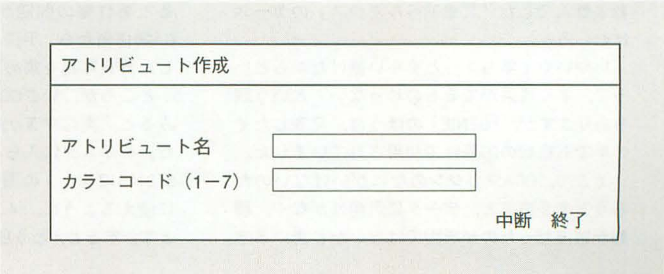


図2　アトリビュート入力



面を作成すると，その面がそのアトリビュートになるわけです。ついでに解説すると，アトリビュート名の変更や表示カラーの変更は，「Misc」の「Edit Attr...」で行います。

同様に，ほとんど似た操作で，ファイル名とオブジェクト名の指定ができます。CADの場合，セーブするときにファイル名を入力しますが，MODELでは，名前をつけてセーブということはできず，あらかじめ「File」の「New Object...」でファイル名を入力しておきます。

このとき，CADでは必ずファイル名とオブジェクト名が一致していますが，MODELでは別々の名前をつけることができます。

面の作成

ここがCADと最も違うところです。CADでは，左クリックで水平方向，右クリックで垂直方向に動かし，スペースで点を確定し，リターンで面を確定していました。これが MODELでは，

右クリック：十字カーソルの移動

左クリック：点の確定

左のダブルクリック：面の確定

となります。では，水平方向，垂直方向に動かせないのかといえば，もちろんできます。

シフト＋右クリック：水平，垂直方向の移動

という形になります。

とにかく，この独特のマウス操作に慣れることが，MODEL攻略の最大のポイントです。とにかく，練習しましょう。でも，よく考えると，こちらの操作のほうが一般的で，CADのほうがむしろ独特ですよね。

ほかにも面を作成するときに頻繁に使用する機能として，「最近点」と「平面投射」がありましたが，当然MODELでも可能です。これは，すべてキーボードで操作します。

「V」：最近点

「P」：平面投射

ただ，CADの平面投射は，3面図の中の点線で囲まれる図面（ポイント図面？）に垂直に投影しますが，MODELにはこのポイント図面という概念が特にありません。ですから，現在マウスカーソルがある図面に垂直に投影します。平面投射を行う場合，マウスカーソルの位置に気をつけてください。

また，MODELは，「CLR」キーを押すことで，随時数値入力状態になります。ノギスで測ってモデリングするような方には，便利な機能でしょう。

面選択指定の基本

以上のように，CADでできることはだいたいMODELでもできることがわかったと思います。しかし，それだけではMODELを使うメリットはありません。ではそろそろ，MODELならではの機能を紹介していきましょう。

CADでは，一度作成された面に対して編集するという機能はほとんどありませんでした。複数の面の削除ですら，「面呼び出し」の機能でポイント面を移動して，1面1面削除しなければいけません。それに対してMODELでは，複数の面の集合を指定し，一度にいろんな効果を

復活のCGAマガジン（前編）

というわけで，長い沈黙を破ってCGAマガジンが近々発行されます。なぜ，いまごろ？ という前に，そもそもなぜCGAマガジンは休刊していたのでしょうか。理由は大きく2つあります。まずひとつは，あまりに手間がかかりすぎ，スタッフの誰もやりたがらないという理由です（わかりやすい）。もうひとつ最大の理由は，その存在理由がよくわからなくなったからです。

もともとCGAマガジンは，データベースを共有し，CGA作品制作の負担を軽減するという意図がありました。しかし，1年ちょっと発行したものの，CGAコンテストにおいても，CGAマガジンのデータを流用した作品はひとつも発表されませんでした（「冥皇龍ベルギウス」のケースはむしろ逆）。

“いやいや1年ちょっとぐらい続けたからといって，すぐ成果がでるものじゃない”という説もありますが，「GENIE」のほうは，発表したその年でも複数の応募作で利用されていました。

そこで，CGAマガジンのなにがいけないのだろうと考え直すと，データに汎用性がなく，種類が散漫だったのが原因ではないかと思います。たとえば，エンジンなど，技術的にはそれ自体は面白いデータであっても，CGA作品にそのデータを使う機会は非常に限られています。また，戦闘機など，精密に作られていても，1機だけでは作品になりません。もっと利用価値があるデータを，系統だてて提供する必要があるように感じました。

しかし，それを実行するのはなかなかたいへんです。そこでとりあえず考えたのは“背景画像データ集”です。コンテストを見ても背景に手を抜いた作品はよくあります。みなさん背景画像に困っている証拠でしょう。手で描くのは結構たいへんですし，風景写真をスキャナで取ると著作権の問題が発生します。そこで，DōGA関係者から，手描きの背景画像や自分で撮影した背景写真を集めました。

ところが，いざCGAマガジンとして編集してみると，実にマヌケな事態に陥ってしまいました。ディスクに入らないのです。背景画像は，512×512ドットの画像1枚ではなく，BGMAKEに使えるように，4，5枚がセットになっています。すると，どう圧縮しても2HDのFD2枚には，ほんの数セットしか入りません。その結果，いろいろ苦労して背景画像のデータベースはできたものの，発行する手段がなくなってしまったのです。そんなこんなで，CGAマガジンは再び袋小路に入ってしまいました。

しかし，ふとした理由で，CGAマガジンを発行しなければいけない事態に陥りました。その理由とは……，恥ずかしくて書けません。発行されるCGAマガジンをご覧ください。“もしかして，あの件かな？”とピンとくる読者の方もいらっしゃるでしょう。うっ，なさけない。

というわけで，休刊になった原因はほとんど未解決のまま，CGAマガジンは急きょ1号だけ追加されることになりました。それならちょうどいいやということで，最新のツール類もまとめて発表します（本当は，本誌の12月号の付録ディスクに入れてほしかったんですけど，もういっぱいで入らないということで……）。

CGAマガジンの具体的な内容については，12月号でお知らせします。また，背景画像データ集も，別のかたちで発表しますので，詳しくは「各読者連絡事項」をご覧ください。

加えることができます。そこでまず，複数の面を指定する方法について解説します。

試しに，多数のポリゴンがある物体をロードしたあとで，「Select」の「Select Area」を選択してください。画面最上部のメニューの横の表示が「Polygon」から「Select」に変わります。なお，この「Select Area」はキーボードの小文字の「s」でも選択できます。頻繁に使うことになると思いますので，覚えておいたほうが楽でしょう。

この状態で，十字カーソルを動かし，どれか１つの図面上で(透視図も可能)，対角にあたる２頂点をクリックして，長方形領域を指定します(写真２)。すると，その長方形領域に含まれる頂点をもつ面のすべてが選択状態になり，虹色に(？)点滅します。たとえば，この状態で「DEL」キーを押せば，選択された面がすべて削除されます。どうです，簡単でしょう。

しかし，これだけではちょっと複雑な物体に対して，特定の面だけ選び出すのは難しいと思うかもしれませんが，安心してください。この面の指定方法に関する機能は結構用意されています。

まずは，基本事項から。

１） 辺が含まれてもダメ

図３のように，指定した長方形領域に，辺が掛かっていても，頂点が含まれない場合，選択されません。

２） 頂点が１つでもＯＫ

その面のすべての頂点が長方形領域に入っていなくても，どれか１頂点でも含まれれば選択されます。図３の面Ｃもそのケースといえます。

ただし，これは，次のように変更可能です。

３） 「Setup...」で全頂点

「Misc」の「Setup...」で，図４のようなウインドウが表示されます。この「選択モード」を，「一部」から「全体」に切り替えてください。するとこんどは図５のように，その面のすべての頂点が長方形領域に入っていなければ，選択されないようになります。

なお，この「一部」と「全体」の切り替えは，メニューをいちいち選択しなくても，キーボードの「ｃ」ボタンで切り替わります。そうすると，現在「一部」なのか「全体」なのかわからなくなってしまいそうですが，長方形領域指定時，１点目を確定したあと，画面最上部に「Select Area 一部」というように表示されていますので，確認するようにしてください。

４） 「Deselect Area」で選択解除

すでに複数の面が選択されている状態で，「Select」の「Deselect Area」を選んでください。「Deselect Area」は，長方形領域で指定される範囲に，選択状態の面があった場合，選択状態を解除します。

以上が基礎的な面の選択法です。理解できましたか？

これらの組み合わせでかなり複雑な物体でも，特定の面だけを選択できます。では，ここで問題です。

演習問題

問題：図６のような形状の正八角柱の側面部分に相当する面だけを選択せよ。

まず考えるのが，「Select Area」で図７のような範囲を指定する方法でしょう。しかし，よく見ると，これでは長方形領域に頂点は含まれていないので，１面も選択できません。

また，側面の頂点と底面や上面の頂点は共有されているため，「Select Area」で角柱全体を長方形領域で囲んでしまうと，必ず底面や上面も選択されてしまいます。

そこでまず，「Select Area」の「一部」で，図８のような範囲を指定します。これで，すべての側面と上面が選択されます(底面は選択されません)。

次に，「Deselect Area」に切り替え(小文字の「d」)，

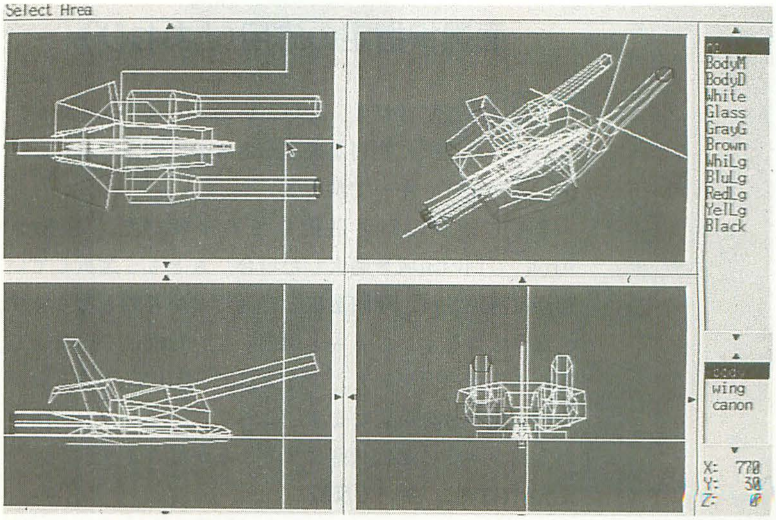


写真２　対角にあたる２頂点を指定

図３　面Ａ，Ｂは選択されず，面Ｃだけ選択される

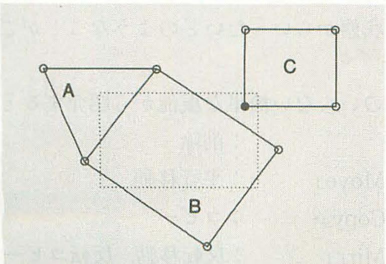


図４　システム設定

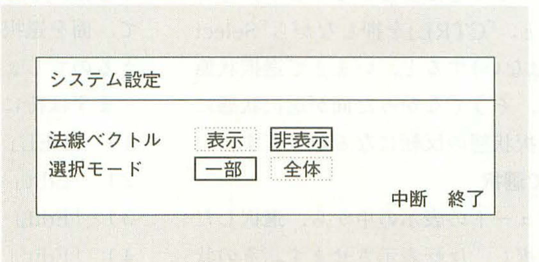


図５　面Ａは選択されるが，面Ｂはされない

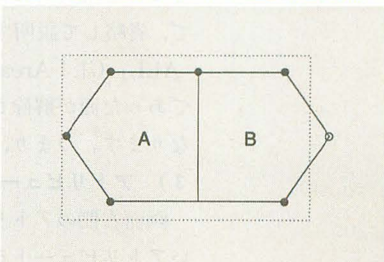


図6　演習問題の図形

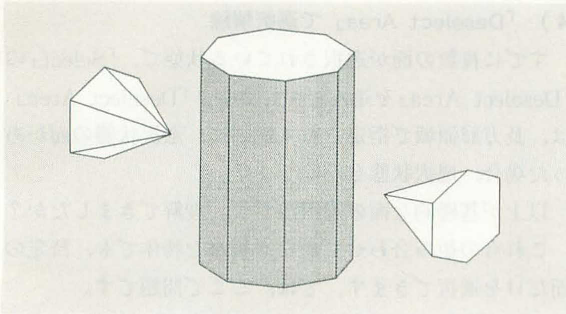


図7　誤った指定

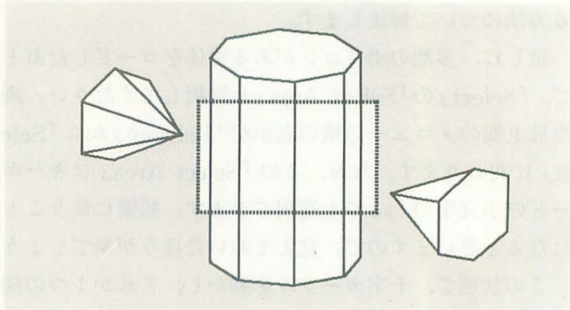


図8　側面と上面を選択

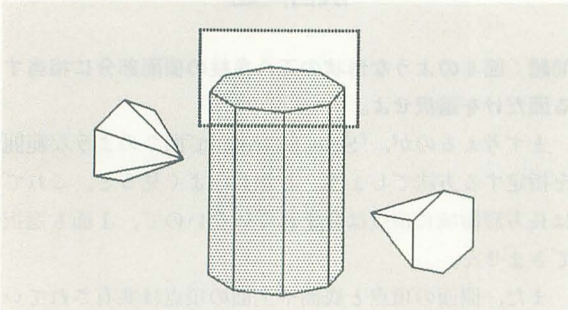
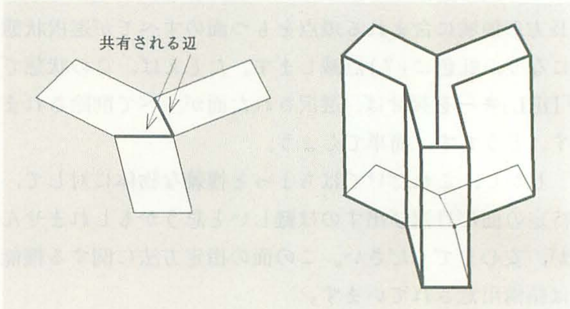


図9　共有する辺は立ち上げない



また「全体」に切り替え(「c」)，再び図9とまったく同じ範囲を指定します。すると今度は「全体」なので，選択解除の対象となるのは，上面の正八角形だけになります。こうして，側面だけを選択状態にすることができました。

面選択指定の応用(とおわび)

上記の基礎だけでも，だいたいのことはできますが，ほかにも面選択に便利な機能があるので紹介します。

1）「SHIFT」キーで加える

すでに複数の面を選択してある状態で，「Select Area」で別の領域を選択した場合，もともと選択していた面は解除され，今回選択した面だけが有効となります。

そうではなく，もともと選択していた面はそのままで，それにほかの面を加えたいときは，「SHIFT」を押しながら長方形領域を指定します。これで，バラバラな場所にある面を同時に選択状態にすることができます。

2）「CTRL」キーで反転

同様に，「CTRL」を押しながら面を選択すると，XORを取ります。XOR……意味がわかりませんし，わかったところで現実的に使用することは少ないでしょう。そこで，省略して説明すると，「CTRL」を押しながら「Select ALL」(注：Areaではない)すると，いままで選択状態であった面が解除され，そうでなかった面が選択状態になります。つまり，選択状態の反転になるわけです。

3）アトリビュートで選択

画面右側のアトリビュートの表示の中から，選択したいアトリビュートを選択し，反転表示させます。その状態で，「Select」の「Select Attr」を実行すると，そのアトリビュートの面が選択状態になります。上記の「SHIFT」を併用すると，複数のアトリビュートを同時に選択状態にすることもできます。

4）おわび

「Select Attr」があるなら「Deselect Attr」，つまり，現在選択状態にある面のなかから特定のアトリビュートの面だけ選択を解除するという機能があるはずです……がありません。

また，「CTRL」を押しながら「Select Attr」はバグっているようで，動作が正しくありません。

そして，CADでいう「面呼び出し」に相当する「ROLL UP」と「ROLL DOWN」(または「Select Next」と「Select Prev」)はバグというより，根本的にプログラマが勘違いしているようです。ぜんぜん使いものになりません。

以上，おわびとご注意でした。

選択状態から可能なこと

CADと違って，面の選択方法が実に多彩で強力であることは，十分ご理解いただけたと思いますが，そうやって，面を選択した状態からいったいどのようなことができるのでしょうか。

まずは特に説明のいらない簡単な機能から紹介すると，

1）「DEL」キー　　：削除
2）「Edit」の「Move」　　：平行移動
3）「Edit」の「Copy」　　：コピー
4）「Edit」の「Mirr」　　：反転移動，反転コピー

5）「Edit」の「Change Attr」：アトリビュートの変更
6）「Edit」の「Shade」　：SHADE.X 相当
7）「Edit」の「Unshade」　：法線ベクトルを抹消
8）「File」の「SaveSelects」：選択面のみ出力
などがあります。「Shade」なんかは，なかなか便利です。

そのほかで知っておくと便利な機能としては，

9）　必要なものだけvisible

「Misc」の「Invisible」は，現在選択状態にある面を残して，ほかの面を表示しない機能です。単に表示しないだけでなく，以後「Visible」で，すべての面を復活するまで，存在しないものとして扱いますので，選択の対象にも最近点の対象にもなりません。

線が多くなって見にくいときに，とりあえず関係のないあたりを全部非表示にしたり，現在選択状態にある面がどこか確認したり，なにかと使い途が多い機能です。

また，ひとつのファイルに複数の形状が存在している場合など，この機能を利用して，編集しない形状を非表示にしておかないと，邪魔で仕方がありません。

10）　グリッドコピーで並べる

指定された数だけ連続してコピーするのですが，このとき3方向まで指定できます。これだけでは意味がわかりませんね。

1方向の場合，たとえばビルの1階部分だけを作成し，それをZ軸方向に数回コピーして，数階建てのビルにするという使い方になります。2方向の場合，稲の形状をX，Y方向にそれぞれ10回コピーすれば，稲が10×10の100本ならんだ水田になります。そして，3方向の場合，X，Y，Z方向に空間的に並べることができます。

なお，方向は，きっちりとX，Y，Z軸方向にする必要はありません。たとえば，ミカンの木をX方向と，YとZの中間的な方向にコピーすれば，山の斜面のミカン畑になるわけです。

操作方法に少し注意が必要です。最初，コピーと同様に始点と終点を指定することで，最初の方向を指定します。そこで，左をダブルクリックすれば，1方向のみとみなされ，回数入力になります。ただの左クリックでは，十字カーソルが自動的に始点に戻り2つめの方向の終点を入力する状態になります。3方向も同様です。

11）　面立ち上げはアウトライン

CADの角柱作成に相当するのが，「Create」の「面立ち上げ」です。ただ，この面立ち上げは，角柱作成と違って複数の面を同時に立ち上げることができます。

ただ，少し賢いのは，図9のように，辺を共有する複数の面が底面となる場合，その共有する辺だけは立ち上げを行いません。アウトライン部分だけが立ち上がります。こうすることで，たとえば，立体文字の作成など，複雑な形状を立ち上げるとき，余計な面が生成されなくなります。

最後に

以上，MODELについて解説してきましたが，これだけ機能が充実しているなら使ってみようという気になっていただければ幸いです。

次回は，MODELの使い方をもう少し追求するとともに，ほかのツールでめぼしいものを紹介しましょう。

各読者連絡事項

背景画像データ集

別コラムにあるように，CGAマガジンの別冊「背景画像データ集」を希望者に配布します。

内容は，CGA作品の背景として使えそうな画像が，ただぎっしり詰まっているだけです。BGMAKE用に複数でひとつの背景をなしているものもあれば，IC用に単一で使えるものもあります。手描きのものもあれば，私が海外で撮影した写真をもとに合成したものもあります。単なる風景もあれば，心理描写用の背景もあります。要するにいろいろあります。

まだ，整理ができていないのですが，一部を「GraphicGallery」で紹介しておきます。

配布価格は未定です。というか，単にディスクに画像データがたくさん詰まっているだけのものですから，特にお金をいただくものではないでしょう。自分たちだけでもっていてももったいないので，皆様にも利用いただければ幸いです。おそらく，メディア代＋郵送費＋コピーの手間賃（カンパ？）程度になるでしょう。

ディスクの場合，圧縮しても5枚になるか10枚になるか，それ以上になるかもまだわかりません。とりあえず，希望者の数と，メディアはなにがよいかを先に調べたいと思いますので，以下の要領で申し込んでください。

申し込み方法：ハガキ
記入事項：
1）　背景画像データ集希望
2）　希望メディア
（5″2HD，3.5″2HD，MO（128Mバイト）から選択）
宛先：〒533
大阪市東淀川区淡路5-17-2 102号
DōGA内「背景画像データ集希望」係
締め切り：11月30日必着

CGAコンテスト作品募集

Oh!Xの読者の皆さんに告知するのもなんですが，念のため。

第8回アマチュアCGAコンテストの作品を募集します。このコンテストは，アマチュアCGA作品の発表の場を設け，広く一般にPRするとともに，その質的向上を促進するという目的で開催されています。

募集作品：パソコンで制作されたアマチュアCGA作品。使用機種，使用ソフトは問いません。
締め切り：1995年12月31日必着
発表上映会：近畿地区　3月24日
尼崎アルカイックオクトホール
関東地区　4月7日
中野区　中野ZERO

今年は，昨年グランプリの森山さんが作品を作っていないとか，KMCも低調だとか，宇宙人森山さんが密かに制作していた作品は，コンテストのオープニングだったとか，常連の方々のあまり芳しくない噂を耳にしています。もしかして，今年は穴場かも。とにかく，皆さん，作品を出品してください。

(で)のショートプロぱーてぃ——その74

ボケたらツッこまなあかんがな!

Komura Satoshi 古村 聡

今月はボケとツッこみが奏でる(?)お笑いソフトと,すべてのファイル名を大文字にしてくれるツールの2本です。(で)氏の電撃結婚に驚きつつ,楽しんで活用してください。



illustration:T.Takahashi

まいどー,(で)でございますっ!

えーと,まったく個人的なお話してもうしわけないんですが,結婚しましたっ。でへへへ。でもって,現在,あと2時間で空港に行かなきゃいけないんですう。新婚旅行です,新婚旅行。常夏のサンゴ礁,青い海,白い砂,南太平洋のフィジーが待っているのだあ。

でね,式の日は,読者のみなさんも知っていると思いますが,この秋最大の台風が東京へ上陸した日だったのだ。大雨大風大嵐で地下鉄は水浸しだわ,飛行機は飛んでこなくて親戚がこれないわ,その名のとおり,嵐の結婚式。実をいうと,ほんとは新婚旅行は南太平洋でもタヒチの予定だったんですよ。2人ともずーっと「新婚旅行はタヒチだねっ」なんていってあこがれまくって,資料集めるわ,ビデオ買ってくるわで心はタヒチ,タヒチ,タヒチぃ〜。ところが,タヒチではフランスさんが核実験なんかやってくれちゃうわ,おかげで暴動が起こって空港もホテルも破壊されて,ツアーがキャンセルになってしまったとゆー……。

私がいったいなにをしたあ!(もっとも,タヒチ行きの飛行機は台風上陸のためフライト中止になってしまったそうなんですけどね)。うーむ,人間万事塞翁が馬ってやつか。スピーチする人する人,みんながこれをいうんで笑ってしまいましたけど。うむ,これだけムチャクチャだとこれしかいいようないもんな。

ぼけぼけでほげほげ〜

それではプログラムにまいりましょう。おめでたい席にはやっぱり余興ですね。今月1本目のプログラムは漫才番組新人候補シミュレータプログラム(ちと長いな)。X-BASIC用でBOKE.BASとMANZAI.BASです。どうぞっ!

BOKE.BAS&MANZAI.BAS for X680x0
(X-BASIC)
大阪府 周東正男

このプログラムは2本で1本のプログラムです。MANZAI.BASを実行するためには,下準備のためのプログラム,BOKE.BASを実行しなければなりません。

X-BASICを立ち上げて,最初にリスト1を打ち込んでください。リストに間違いがないことを確認して,BOKE.BASという名前でセーブしてください。次に,RUNしてから,FILESコマンドでBOKE.BASがディスクにできていることを確認して,

RUN
NEW

と続けて打ち込み,リスト1を消してください(なお,このBOKE.BASは初期データを作成するためのものですので,実行するのは最初の一度だけでOKです)。

続いて本体プログラムであるリスト2を打ち込んで,MANZAI.BASという名前でセーブしてください。ここまで終わったら,おもむろにRUN。

このプログラムには長い長〜いストーリーがあるんですが省略してちゃっちゃかゲームの遊び方を説明しましょう(周東さんごめんね〜)。漫才師って「ボケ」と「ツッこみ」の2人がいますよね。この「ボケたらツッこまなあかんがな!」は,そのうちの「ボケ」を演じるゲームです。

スタート画面でリターンキーを押すと,あなたのコンビの名前を聞いてきますので,コンビ名,ボケの名前,ツッこみの名前をそれぞれ入力してください。ゲームがスタートしますと,まず,相方のツッこみがなにやらしゃべるセリフが表示されます。たとえば,

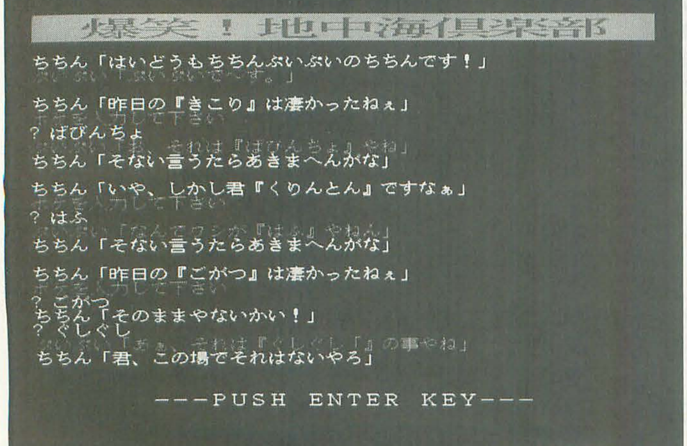
つっこみ「それにしてもお前「らんぼー」みたいやなぁ」

というふうに。そうしたら,ボケであるあなたは,「」で囲まれた単語に似た(似てなくてもいいんですけど)面白い単語を,全角ひらがなで入れてください(漢字に変換してはいけません)。ここで,

こんどー

と入れてみます。すると,

ボケ「誰が「こんどー」やねん」
ツッこみ「なにいうてんねん」



BOKE.BAS&MANZAI.BAS

とボケのセリフが表示されます。

これを3回繰り返すと，審査員から点数が表示されます(テレビ番組の「お笑いスター誕生」みたいなもんなのだな。あ，ちなみに「爆笑地中海倶楽部」ってのがゲーム中の番組のタイトルなのね)。どんどんボケまくって高得点を目指してください。最後にいま演じた漫才をファイルに落とすのなら“Roll Up”，落とさないなら“Roll Down”キーを押すとスタート画面に戻ります。スタート画面で“ESC”キーを押すとゲーム終了です。

きみとはやっとれんわ。

しつれいしましたー。

わー，すごいすごいすごい。大阪ってすごいな。そういや，編集室にも大阪出身の人がいるんですけどね。もー，この人たちの会話ってば，天然の漫才師ってーくらい，ボケまくりツッコミまくってるな，と思っていたんです。そしたらこういうソフトが開発されていたのか。恐るべし大阪人。そういえば，大阪人は小学校に漫才の授業があって，ボケとツッコミクラスに分かれて特訓するとか聞いたことがあるぞ(あらへんあらへん)。うーむ，テキストゲームでここまで面白いとはねえ。恐れいりました。テキストゲームだけに素敵とゲーム，なんてね。え，座布団全部取れって，そんなあ(番組が違う気もするぞ)。

全部大文字XCASE.X

さて，今月2本目のプログラムは，すべてのファイルを大文字ファイルにするXCASE.Xです。どうぞっ。

XCASE.X for X680x0

(要Cコンパイラ，CASE.X)

千葉県　荒木研二

このプログラムがなんのためにあるかというと，簡単にいうとMS-DOS対策です。Human68kではファイル名に小文字を使うことができますが，MS-DOSでは小文字を使うことができません。で，先ほど，パスの通っているディレクトリにあることを確認した，CASE.Xというコマンドを使えば，ファイル名を大文字にすることができるのです(詳しくはHumanのマニュアルを見てください)。たとえばMOやZIPディスクのような容量の大きいディスクで，X68000からMS-DOS系のマシンにデータを渡そうとしたときに，ディレクトリのそこかしこに小文字のディレクトリやファイルが散らばっている場合があります。そんなとき，いちいち手作業でCASE.Xを使いファイル名を大文字にするのも大変ですよね。そんなときに役立つのがこのプログラムです。

このプログラムはCコンパイラ用に書かれています。リスト3をエディタで打ち込んで，XCASE.Cの名前でセーブします。それから，Cコンパイラでコンパイルして実行ファイルを作ってください。XC ver.2以降を使うのであれば

A>CC /Gs30000 XCASE.C

でコンパイルできます。

実行ファイルXCASE.Xができたら，使い方です。このプログラムを実行するには，パスの通ったディレクトリに，ファイル名を大文字にするプログラム，CASE.Xがある必要があります。¥BINあたりには，たいていパスが通っているでしょうから，¥BINにCASE.Xがあることを確認してください。その状態で，

A>XCASE

とコマンドラインから打ち込みプログラムを実行させると，カレントディレクトリより下にあるすべてのディレクトリの名前と，そのディレクトリの中にあるファイルの名前をすべて大文字に変えます。引数はありません。

ということですが，こ，このプログラムってば，ファイル名を大文字にするために外部コマンドを使っているのですねっ。うーむ，妙に短いプログラムだと思ったら。

しかし，せっかく外部コマンドを使うんだったら，CASE.Xだけじゃなく，好きなコマンドを使えるようにしてくれればよかったような気もしますな。たとえば，PRINTコマンドを使えば全部のディレクトリの中にある.DOCのファイルだけプリントできるし。

あ，そうか，PRINT.XをCASE.Xと名前を変えてしまえばいいのか。ちなみに，

A>COPY PRINT.X CASE.X

以上のように打ち込むと，ああっ，本物のCASE.Xが消えてしまった！(くれぐれも真似しないよーに)。

ところでWindows95になると，この小文字が使えないっていう制約はどうなるんでしょうかね。確か，小文字はおろか，ファイルネームも255文字まで使えるようになるそうなんで，なにもしなくてもすむんでしょうか。どなたか詳しい方教えてくださいまし。Windows95ではどの機種でもデバイスドライバなしでIBM形式のMOなんかも読み書きできるそうですしね。

ということで今月はおしまい。

そろそろ空港に行く時間になりました。南の海が青い海が白い砂が待ってるぜ，べいべー。それではまた来月っ！



リスト1 BOKE.BAS

```
10 /*
20 /*  データファイル作成用プログラム
30 /*
40 /*                              Presented by MASAO.S
50 /*
60 cls : console ,,0 : color [,,rgb(31,0,0)+1,rgb(31,31,31)+1]
70 /*
80 /*  初期設定
90 /*
100 int fp,i
110 str ji1,ji2,conb,boke,tukk
120 str neta(9)[14]={
130   "かんびーる","えんだか","びすまるく","くりんとん",
140   "こうむいん","ふぁんた","すりがらす","ごがつ",
150   "きこり",    "えすえふ"
160   }
170 str datt(9)[12]={
180   "かんがるー","わるいんだか","ですますく","くりきんとん",
190   "ほーむいん","あんた","たびがらす","ごがつみどり",
200   "かっこいい","えすえす"
210   }
220 /*  色々な名前決定エリア
230 print : color 2
240 print "カナや漢字や半角でもお好きにどうぞ！"
250 print : color 3
260 print "コンビ名を入れて下さい"; : input conb
270 print "ツッこみの人の名前を入れて下さい"; : input tukk
280 print "ボケの人の名前を入れて下さい"; : input boke
290 /*
300 /*  例：コンビ名＞エンドレス・サブ  ツッこみ＞鮫島  ボケ＞まんむぅ
310 /*
320 fp=fopen("NAMAE.DAT","c")
330         fwrites(conb+chr$(&HD)+chr$(&HA),fp)
340         fwrites(tukk+chr$(&HD)+chr$(&HA),fp)
350         fwrites(boke+chr$(&HD)+chr$(&HA),fp)
360 fcloseall()
370 /*
380 /*  元ネタ決定エリア
390 /*
400 fp=fopen("DATT.DAT","c")
410 for i=0 to 9
420         ji1=neta(i) : ji2=datt(i)
430         fwrites(ji1+chr$(&HD)+chr$(&HA),fp)
440         fwrites("1"+chr$(&HD)+chr$(&HA),fp)
450         fwrites(ji2+chr$(&HD)+chr$(&HA),fp)
460         fwrites(    chr$(&HD)+chr$(&HA),fp)
470 next
480 fcloseall()
490 end
```

リスト2 MANZAI.BAS

```
10 /*
20 /* 『ボケたらツッこまなあかんがな』    1995/08/22 AM1:35 完成
30 /*
40 /* 爆笑！地中海倶楽部
50 /*    （毎週＜月＞～＜金＞  ＰＭ５：００～５：４５絶賛放映中）
60 /*
70 /* 発案者 兼 設計者 兼 ネタ帳作成 兼 プログラマー：周東  正男
80 /*
90 screen 1,3,1,1 : console 2,28,0 : locate ,,0
100 randomize(atoi(right$(time$,2)))
110 color[0,rgb(31,15,0)+1,rgb(7,7,31)+1,rgb(31,31,31)+1]
120 /*
130 /*  初期設定
140 /*
150 str hyoka(4)[26]={ "ダウソタウソ","縛笑問題","フォークダンスｄｅ
いろは坂",
160                    "怪砂利木魚", "コントホームラン" }
170 str furi(7)[20]={ "いや、しかし君『","今、世間では『","それにし
てもお前は『","昨日の『",
180                    "』ですなぁ","』らしいですなぁ","』みたいやな
ぁ","』は凄かったねぇ" }
190 str bona(7)[20]={ "お、それは『","あぁ、それは『","えっ？いま君
『","なんでワシが『",
200                    "』やね","』の事やね","』て言うた？","』やねん
" }
210 str tuna(3)[30]={ "お前アホか","なに言うてんねん","そない言うた
らあきまへんがな",
220                    "君、この場でそれはないやろ" }
230 str jis[100]="あいうえおかきくけこさしすせそたちつてとなにぬねの
はひふへほまみむめもや＊ゆ＊よらりるれろわ＊＊＊ん"
240 str jis2[68]="がぎぐげござじずぜぞだぢづでどばびぶべぼぱぴぷぺぽ
っあいうえおやゆよ"
250 str conb,tukk,boke,memo(15)[96],ji[96],neta(9)[96]
260 str rita(9,9)[96],jib,ky,nas(9,4),nas2(33)
270 int ten,kai,fp,i,j,ck,nn,no,no2(2),rn,suu(9)
280 int mm1(33)={1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,5,5,5,5,5,5,5,5,5,3
,0,0,0,0,0,7,7,7}
290 int mm2(33)={0,1,2,3,4,0,1,2,3,4,0,1,2,3,4,0,1,2,3,4,0,1,2,3,4,2
,0,1,2,3,4,0,2,4}
300 for i=0 to 9 : for j=0 to 4 : nas(i,j)=mid$(jis,i*10+j*2+1,2) :
next : next
310 for i=0 to 33 : nas2(i)=mid$(jis2,i*2+1,2) : next
320 /*
330 /*  データ読み込み
340 /*
350 for i=0 to 9 : neta(i)="" : for j=0 to 9 : rita(i,j)="" : next :
 next
360 fp=fopen("NAMAE.DAT","r")
370   freads(conb,fp)
380   freads(tukk,fp)
390   freads(boke,fp)
400 fclose(fp)
410 fp=fopen("DATT.DAT","r")
420 for i=0 to 9
430   for j=0 to 11
440     ck=freads(ji,fp)
450     if ck<1 then break
460     if j=0 then neta(i)=ji
470     if j=1 then suu(i)=atoi(ji)
480     if j>1 then rita(i,j-2)=ji
490   next
500 next
510 fclose(fp)
520 /*
530 /*  メイン・ループ
540 /*
550 while 1
560   title()
570   fill(0,0,511,31,rgb(2,25,2)+1)
580   symbol(42,5,"爆笑！地中海倶楽部",2,1,2,rgb(20,2,2)+1,0)
590   for i=0 to 15 : memo(i)="" : next
600   ten=0 : kai=0 : sikai(0,0) : nn=2
610   color[,,rgb(31,10,10)+1]
620   ji=tukk+"「はいどうも"+conb+"の"+tukk+"です！」"
630   print ji : memo(0)=ji : color 2
640   ji=boke+"「"+boke+"で～す。」"
650   print ji : memo(1)=ji : color 3
660   print : for i=0 to 2 : no2(i)=0 : next
670 /*
680 /*  ゲーム開始
690 /*
700   while kai<3
710     while 1
720       no=int(rnd()*10)
730       if kai<1 then break
740       if kai=1 then if no2(0)<>no then break
750       if kai=2 then if no2(0)<>no and no2(1)<>no then break
760     endwhile
770     no2(kai)=no : kai=kai+1 : rn=int(rnd()*4)
780     ji=tukk+"「"+furi(rn)+neta(no)+furi(rn+4)+"」"
790     rial()
800     color 2 : print "ボケを入力して下さい" : color 3 : jib=""
810     while jib=""
820       input jib
830       if jib="" then {
840       print tukk;"「何？よう聞こえへん。もう一回いってみぃ。」"
850                     }
860       if jib=neta(no) then {
870         print tukk;"「そのままやないかい！」
880         jib=""
890                          }
900     endwhile
910     rita(no,suu(no))=jib : suu(no)=suu(no)+1
920     if suu(no)>9 then suu(i)=0
930     rn=int(rnd()*4) : color 2
940     ji=boke+"「"+bona(rn)+jib+bona(rn+4)+"」"
950     rial() : color 3
960     ji=tukk+"「"+tuna(rnd()*4)+"」" : rial() : print
970     ten=ten+black(neta(no),jib)
980   endwhile
990   botton()
1000  color[,,rgb(7,7,31)+1] : print
1010  ji=tukk+"「君とはやっとられんわ」" : rial()
1020  ji=conb+"「どうもありがとうございました」" : rial() : print
1030  ten=int( ten / 3 )
1040  sikai(1,ten)
1050  kioku()
1060 endwhile
1070 /*
1080 /*  終了
1090 /*
1100 end
1110 /*
1120 /*  声援
1130 /*
1140 func koe()
1150   print : color 1 : print "ワー！！"
1160   print "バチバチバチ・・・"
1170   print : color 3
1180   mati(200)
1190 endfunc
1200 /*
1210 /*  司会語録
1220 /*
1230 func sikai(dig;int,ten;int)
1240   int le
1250   if dig=0 then {
1260   symbol(125,450,"注：ローマ字・ひらがな・全角キーを押して下さい
",1,1,0,1985,0)
1270   koe() : color 6
1280   print "はい、いつもの「おお笑い海水浴場」のコーナーにやってき
ました。"
1290   koe() : color 6 : mati(200)
1300   print "それでは、はりきってどうぞ！！"
1310   koe() : color 3 : print
1320   botton() : print
1330   symbol(125,450,"注：ローマ字・ひらがな・全角キーを押して下さい
",1,1,0,0,0)
1340                  }
1350   if dig=1 then {
1360     koe() : color 6
1370     le=4-int(ten/20) : if le<0 then le=0
1380     print "はい、どうもありがとうございました。"
1390     print "さぁ、審査結果をどうぞ！！" : color 3 : botton()
1400     symbol(8-96*(ten<>100)-95*(ten<10),30,itoa(ten),28,35,0,rgb(
29,0,0)+1,0)
1410     mati(500): koe() : color 6
1420     symbol(8-96*(ten<>100)-95*(ten<10),30,itoa(ten),28,35,0,0,0)
1430     symbol(380-20*(ten<>100)-20*(ten<10),35,itoa(ten),7,9,0,rgb(
29,0,0)+1,0)
1440     print "・・・";ten;"点です！！！"
1450     print "これは将来の";hyoka(le);"かな？" : mati(200)
1460     print "それではまた次回まで！さようなら～！！"
1470     color 3 : botton()
1480                  }
```

▶「人類が『2』を発見したとき，すべての文明はここから始まった」。イギリスの哲学者の言葉です。たまにはこういう本を読むのも面白いかもしれませんよ。特に不眠症の方にお勧め。
太田 志輝(19)北海道

```
1490 endfunc
1500 /*
1510 /*  ボタン処理
1520 /*
1530 func botton()
1540   print
1550   print space$(12);
1560   print "---PUSH  ENTER  KEY---"
1570   print
1580   while inkey$<>chr$(&HD) : endwhile : cls
1590 endfunc
1600 /*
1610 /*  ウェイト
1620 /*
1630 func mati(ti;int)
1640   int i
1650   for i=0 to ti*10 : next
1660 endfunc
1670 /*
1680 /*  タイトル
1690 /*
1700 func title()
1710   int i,flag=0
1720   str ky=""
1730   fill(0,0,511,511,0)
1740   for i=0 to 1
1750   box(0+i*2,0+i*2,509+i*2,509+i*2,rgb(31,31,31)+1)
1760   symbol(25+i*4,15,"ボケたら",3,3,2,rgb(21+i*10,21+i*10,0)+1,0)
1770   symbol(105-i*4,90,"ツッこまな",3,3,2,rgb(0,0,20+i*11)+1,0)
1780   symbol(15-i*4,175-i*4,"あかんがな",4,5,2,rgb(25+i*6,25+i*6,31)+1,0)
1790   next
1800   symbol(165,400,"Push Enter Key",1,1,2,rgb(31,31,31)+1,0)
1810   symbol(365,470,"ESCで終了します",1,1,1,rgb(25,5,5)+1,0)
1820   symbol(240,490,"カンガルーを考える会代表<<Para>>",1,1,1,rgb(0,25,0)+1,0)
1830   while ky<>chr$(&HD)
1840     ky=inkey$ : if ky=chr$(&H1B) then flag=1 : break
1850   endwhile
1860   fill(0,0,511,511,0)
1870   if flag=1 then owari() : end
1880 endfunc
1890 /*
1900 /*  メモリアル関数
1910 /*
1920 func rial()
1930   memo(nn)=ji : nn=nn+1
1940   print ji
1950 endfunc
1960 /*
1970 /*  メモリアル本番
1980 /*
1990 func kioku()
2000   int flag
2010   str nam,en
2020   locate 0,10 : print "今日の漫才をファイルに落としますか"
2030   print space$(15);" ( Y< Roll-Up > / N< Roll-Down > ) "
2040   ky=""
2050   while ky=""
2060     ky=inkey$
2070     if ky=chr$(&HE) then flag=1 : break
2080     if ky<>"" then break
2090   endwhile
2100   if flag=1 then {
2110     nam="" : en=""
2120     while nam=""
2130       print : print "ファイル名を入力して下さい"
2140       input "(拡張子は入れなくていいです)";nam
2150     endwhile : print
2160     while en=""
2170       input "演目名を入れて下さい";en
2180     endwhile
2190     fp=fopen(nam+".TXT","c")
2200     ji="演目名:"+en
2210     fwrites(ji+chr$(&HD)+chr$(&HA),fp)
2220     ji=space$(48)+"得点:"+itoa(ten)+"点"
2230     fwrites(ji+chr$(&HD)+chr$(&HA),fp)
2240     for i=0 to nn-1
2250       ji=memo(i)
2260       if i=nn-2 or i=nn-5 or i=nn-8 or i=nn-11 then {
2270         fwrites(   chr$(&HD)+chr$(&HA),fp)
2280                                                          }
2290         fwrites(ji+chr$(&HD)+chr$(&HA),fp)
2300     next
2310     fclose(fp)
2320                   }
2330   cls
2340 endfunc
2350 /*
2360 /*  書き込み
2370 /*
2380 func owari()
2390   fp=fopen("DATT.DAT","c")
2400   for i=0 to 9
2410     ji=neta(i) :      fwrites(ji+chr$(&HD)+chr$(&HA),fp)
2420     ji=itoa(suu(i)) : fwrites(ji+chr$(&HD)+chr$(&HA),fp)
2430     for j=0 to 9
2440       ji=rita(i,j)
2450       if ji<>"" then fwrites(ji+chr$(&HD)+chr$(&HA),fp) else break
2460     next
2470     fwrites(   chr$(&HD)+chr$(&HA),fp)
2480   next
2490   fclose(fp)
2500   print "プロデューサー「貰うもんもらってさっさと帰えんな!」
2510 endfunc
2520 /*
2530 /* 究極のブラックボックス・<<点数計算>>
2540 /*
2550 func black(a;str,b;str)
2560   int c,d,e=0,f,g=0,h=0,s,t,u,o(1,4),p(1,9),v,w
2570   str k,k2
2580   c=strlen(a) : d=strlen(b) : if c>d then e=2
2590   if c=d or c+9<d then e=10
2600   if c<d and c+8>=d then e=20 : for i=0 to 9
2610   if rita(no,i)<>"" then h=h+1 else break : if rita(no,i)=b then g=g+1
2620   next : f=int(20*(100-(g*100/h))/100) : e=e+f
2630   for s=0 to 1
2640     if s=0 then k=a else k=b
2650     for t=0 to (strlen(k)/2)-1 : u=0 : k2=mid$(k,t*2+1,2)
2660     for i=0 to 9 : for j=0 to 4
2670         if k2=nas(i,j) then {
2680           u=1 : o(s,j)=o(s,j)+1 : p(s,i)=p(s,i)+1 : break
2690                               }
2700     next : if u=1 then break
2710     next
2720     if u=0 then {
2730       for i=0 to 33
2740         if k2=nas2(i) then {
2750           o(s,mm2(i))=o(s,mm2(i))+1
2760           p(s,mm1(i))=p(s,mm1(i))+1 : u=1 : break
2770                               }
2780       next
2790                   }
2800     if u=0 then {
2810       if k2="-" or k2="—" then {
2820       k2=mid$(k,(t-1)*2+1,2)
2830       for i=0 to 9 : for j=0 to 4
2840       if k2=nas(i,j) then {
2850         o(s,j)=o(s,j)+1 : p(s,i)=p(s,i)+1
2860         u=1 : break
2870                             }
2880       next : if u=1 then break
2890       next
2900                                       }
2910                   }
2920     next
2930   next
2940   for i=0 to 4:if o(0,i)<=o(1,i) and o(0,i)+4>o(1,i) then v=v+8
2950   next
2960   for j=0 to 9:if p(0,i)<=p(1,j) and p(0,i)+4>p(1,j) then w=w+2
2970   next : e=e+v+w
2980   return(e)
2990 endfunc
3000 /*
3010 /*  打ち込んだあなたへ・・・『あ~りが~とさ~ん!』
3020 /*
```

リスト3 XCASE.X

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <doslib.h>

void casesubdir(char *chdir)
{
  char cmdline[256];
  char casecmd[]="case /d ";
  char wildcard[]="*.*";
  struct FILBUF buffer;
  char wildname[256];

  strcpy(cmdline,casecmd);
  strcat(cmdline,chdir);
  printf("[ %s ] ¥n",cmdline);
  system(cmdline);

  strcpy(wildname,chdir);
  strcat(wildname,"¥¥");
  strcat(wildname,wildcard);
  if(FILES(&buffer,wildname,0x10) >=0){
    if((strcmp(buffer.name,".")!=0 )&&(strcmp(buffer.name,"..") !=0 )){
      strcpy(cmdline,chdir);
      strcat(cmdline,"¥¥");
      strcat(cmdline,buffer.name);
      casesubdir(cmdline);
    }
    while(NFILES(&buffer) >= 0){
      if((strcmp(buffer.name,".")!=0 )&&(strcmp(buffer.name,"..") !=0 )){
        strcpy(cmdline,chdir);
        strcat(cmdline,"¥¥");
        strcat(cmdline,buffer.name);
        casesubdir(cmdline);
      }
    }
  }

}

void main( void )
{
  struct FILBUF buffer;

  printf("[ case /d *.* ]¥n");
  system("case /d *.*");
  if(FILES(&buffer,"*.*",0x10)>=0){
    if((strcmp(buffer.name,".")!=0 )&&(strcmp(buffer.name,"..") !=0 ))
      casesubdir(buffer.name);
    while(NFILES(&buffer)>=0)
      if((strcmp(buffer.name,".")!=0 )&&(strcmp(buffer.name,"..") !=0 ))
        casesubdir(buffer.name);
  }
}
```

▶新機種のために貯金をしてたのにとんでもない出費が……(秘密)。あんまりだ。
松本 健一(23)静岡県

SIDE A

店じまい記念作品(前編):設計方針

Tan Akihiko 丹 明彦

連載2年を過ぎるが，まだまだ理想への道は遠い
今回から2回に分けて完結したプログラムの制作を試みる
まずは，車の挙動やコース設定などの方針を決めていく

突然ながら，諸般の事情により志半ばながら連載を終了しなくてはならなくなった。この機会に，まとめの意味でも完結したプログラムを1本提示すべきであろう。

とはいえ，その内容についてはそれほど欲張れそうもない。理由は2つある。(1)理論面の不備。2年以上連載を続けてきたわけだが，車両挙動に関する理論的考察は中途半端なままである。正直に白状すれば，まだよくわかっていない領域やどう実装すべきか思いつかない領域がたくさん残っているので，大急ぎでそれを解説するわけにもゆかないのである。(2)残り時間の少なさ。具体的には来月の付録ディスクが最後の機会になるのである。

これらのことを考慮して，まことに不本意ながら，今月と来月の2カ月で実現可能な最小限の機能を備えたプログラムにすることに決めた。

ということで，手抜き行為の正当化に終始してしまうことが見えているが，実装の方針を以下に述べていく。優先事項は制作の手間が少ないこと，それなりに見栄えがすること，そして一応はゲームにするということである。

システムの概要

自動車ものとして一般的なレースゲームの雛型を作る。つまり，「サーキットを周回し，ラップタイムを計測する」だけのシステムを作ることにする。

車両挙動

[仕様]

車両挙動は，連載当初で紹介した「走る，曲がる，止まる」という基本中の基本である動きだけを実装する。アクセルを踏めば加速し，ハンドルを切れば旋回し，ブレーキを踏めば減速する。ドリフトはしない。ホイールスピンもしない。グリップ走法のみで，コーナリングはコーナー半径に合わせて速度をコントロールすることだけで行う。

これだけだとつまらなさそうに思えるが，現在流行しているドリフト主体の車ゲームから事実上抹殺されてしまっているドライビングの基本，これに立ち返らせてくれるデザインになっているのである(負け惜しみという話もある)。つまり，コーナー手前では減速し，コーナー脱出では加速するというコーナリングの一連の流れをプレイヤーに要求している。ブレーキングでラップタイムを削り取っていくという地味でストイックなタイムアタックゲームになってしまうが，これはこれで面白さは見出せる。

[実装]

図1に示すような最小限の力学シミュレーションを行う。早い話が高校1年で習うような質点の運動である。加速は以前に実装を済ませたエンジンとトランスミッションのモデル(未公開)からホイールスピン挙動を除いたモデルを使う。車の速度は運動方程式に従う。回転運動は向心力による旋回運動のみとし，車体を剛体とみなした自転運動は計算しないことにする。

一応16ビット機でも動くようにしたいので，あまり複雑な計算はさせない。問題は浮動小数点演算がどの程度重いかということだろう。計算量そのものは少ない(なんだかんだいってもSLASHが働いている時間がいちばん長い)ので，とりあえず浮動小数点で作り，様子を見ることにする。個々の計算は極端に精度を要求するわけではないしオーバーフローすることもないだろうから全部固定小数点でやっても問題はないが，コードの書き換えの手間は馬鹿にならないのだ。FLOAT2.Xのパフォーマンスに期待したいところだ。

サーキット

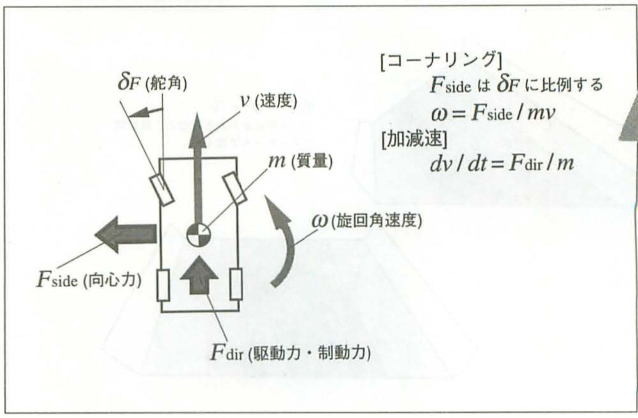
1994年5月号の付録ディスクに載せた高速道路風

サーキットエディタを改造して使う。といってもずいぶん昔の話なので復習しておく。ひとことでいえばサーキットを数種類の部品で表現し，モデリングの手間を省略する手法である（図２）。1本道と限定すれば，サーキットを構成する要素は直線といくつかの曲率を持ったコーナーだけであることがわかる。そこで割り切って，それらを部品にする。部品をどの順番で接続しても矛盾が生じないように外見のデザインも統一する。そして上手に１周分のコースを作ってスタート地点に戻ってくれば完成である。

さて今回走ることになるサーキットであるが，以前のサーキットエディタと比べ，拡張するというよりもむしろレベルを落とす。当時はコースのアップダウンやバンクのついたカーブをサポートしていたが，今回はその両方とも存在しない平坦なサーキットを走るということにする。これは表示上の都合ではなく車両挙動の都合である。サーキットをポリゴンで作る以上，路面は基本的に多面体であり，曲面ではない。この上を走りなんの補正も加えない場合，視界が揺れ動いてとても不快である。路面を疑似的に曲面に見せるにしても，サスペンションの挙動を作って正直に多面体を乗り越えるようにするとしても，処理が複雑になってしまう。この領域に下手に手を出すと収拾がつかなくなり，来月までにまとまらないのが見えている。

コース構造もできるだけ簡素にする。こちらは表示上の都合で，表示を軽くして操作性を上げるためである。コース脇には縁石（コースが平らであるから縁石も平らにする）とグリーンゾーン，それに簡単な壁を配置する。ただ，路面だけは以前よりやや複雑にする。具体的には模様を入れることにする。これだけでスピード感が増すことがわかっているので，非常にお得なのである。

図１　車両挙動（簡易版）



レベル・オブ・ディテールは，サーキットエディタのときに「多段階ディテール」として解説した概念だが，当然今回も用いる（図３）。この部分で手を抜くとSLASHの処理が重くなってしまい，満足なパフォーマンスが出ないのだ。

レベル・オブ・ディテールの考え方は，「３次元コンピュータグラフィックのアプリケーションにおいては，遠くの物体ほど小さく，画面の面積に対して数多く表示される」という事実を利用するものである。小さく表示される遠い物体に高いディテールを

図２　パーツの組み合わせによるコースの構築

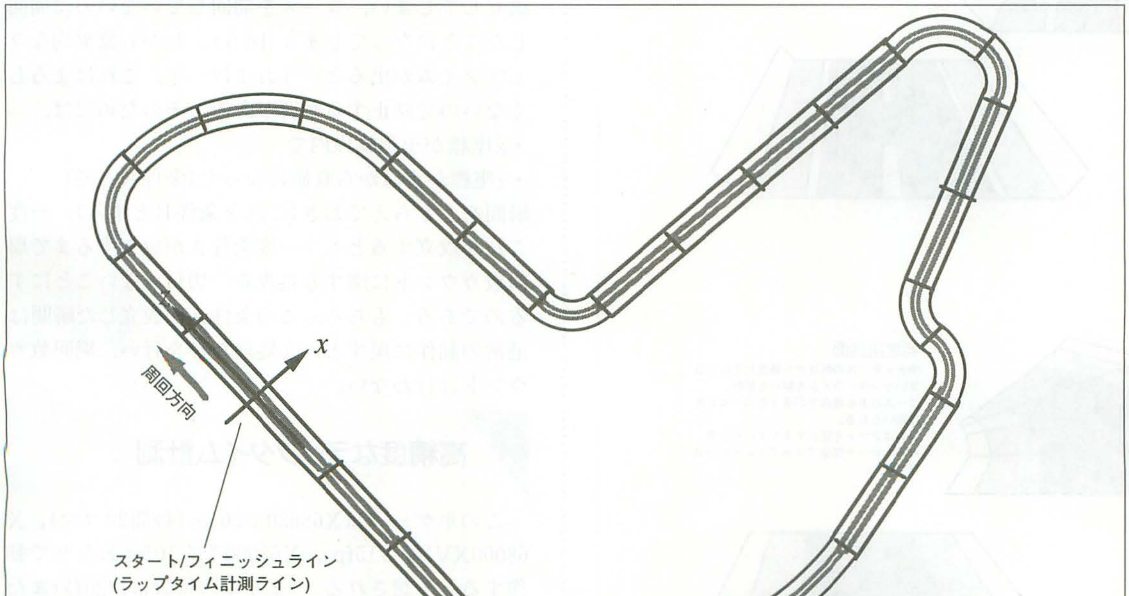
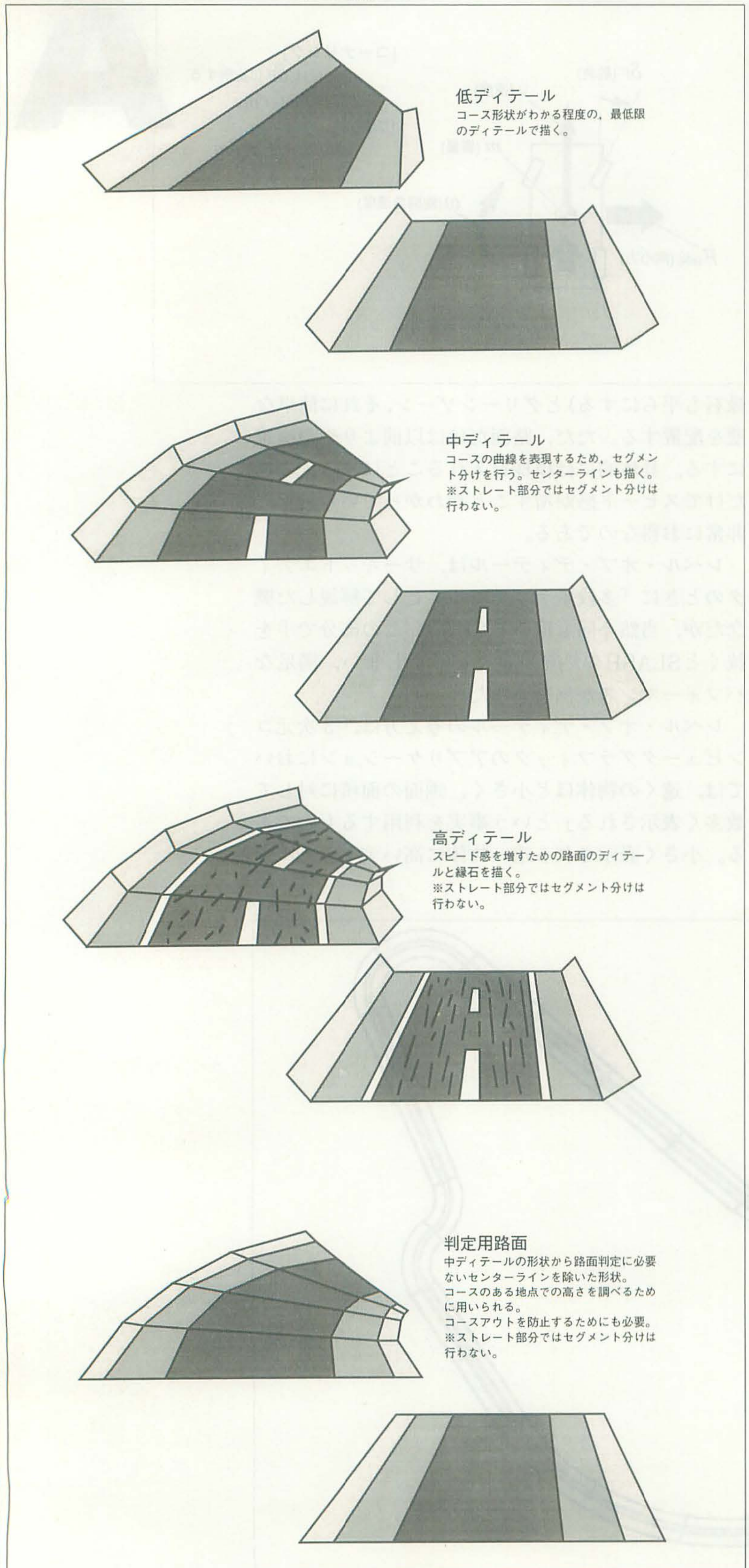


図3　レベル・オブ・ディテール(3段階)



与える必要はない。処理が重くなり，表示も潰れてしまっていいことがない。ディテールを下げれば，それだけ処理も軽くなって平均フレームレートが上がる。または，より遠くの物体まで表示できるようになる。

ディテールはコースの部品ごとに決定する(図4)。どの程度の距離でディテールレベルを切り替えるかは，実際に表示してみて見栄えと負荷を見ながら決めることになるだろう。

裏をかかれない周回数カウント

現実のレースではサーキットを逆に走ったりする奴はいないが，ゲームの世界ではどんなことでも起こりうる。そういう紳士的でない動きをするプレイヤーに対してもシステムは対処しなくてはならない。ありがちなのが周回数カウントにおけるインチキで，プログラムを丁寧に書かないとプレイヤーが計算機の判定の裏をかく方法を見つけ，勝ってもいないのに勝つことができたりする。これは防止しなくてはならない。

もう一度図2を見ていただくと，スタートラインを通過したという判定は，

・x座標が±道幅以内で

・z座標が負値から正値(正確には0以上)になった

瞬間に成立することがわかる(これを条件Aとする)。しかしこれだけだと，スタートラインを通過した瞬間にUターンして一旦スタートラインの後に戻り，もう一度スタートラインを通過すれば，上の条件は成立してしまい，コースを周回していないのに周回したことになってしまう(図5)。しかも驚異的なラップタイムが出るというおまけつき。これはよろしくないので防止する必要がある。そのためには，

・x座標が±道幅以内で

・z座標が正値から負値になった(条件Aと逆)

瞬間をもとらえておき(これを条件Bとする)，一度これが成立するともう一度条件Aが成立するまで周回数カウントに関する処理を一切行わないことにするのである。もちろんこの条件Aが成立した瞬間は通常の動作に戻すという処理だけを行い，周回数カウントは行わない。

高精度なラップタイム計測

この車ゲームはX68030で20fps(秒間20コマ)，X68000XVIなら15fps，X68000なら10fpsあたりで動作すると予想される。つまり車の位置は1/20秒(または1/15秒，1/10秒，以下省略)ごとに求められる。ということは，スタートラインを通過したかどうかの判定は1/20秒ごとに行われる。

ではラップタイムは1/20秒単位でしか求められないのかというとそうではない。図６を見ていただきたい。並走している２台の車は同じフレームにスタートラインを横切ったと判定されるが，ラップタイムが同じであってはならないし，ましてや順位が同じと判定されてもならない。この両者の違いは，スタートラインを通過する前のフレームでの位置と通過したフレームの位置である。これを利用すれば，両者が1/20秒の中のどのくらいの瞬間に通過したかを計算することができる。

今回はホームストレートがｚ軸に平行であると決めているので，位置計算はｚ座標で行う。

ソースコードの公開とSLASHの扱い

この車ゲームは来月の付録ディスクに収録される。ソースコードもつく予定。とはいえ，公開されたのと異なるバージョンのSLASHライブラリを用いて作られているため，コンパイルできない可能性が高い（SLASHは，1994年５月号の付録ディスク「こいのぼりPRO-68K」にて公開されたバージョン2.0以来，一般に出たバージョンは存在しない）。

余裕があればこの車ゲームを公開されたSLASH2.0に対応させたいところだが，バージョンを戻すことになるのであまり気が進まない。それより前向きなのがSLASHの最新版を発表することだが，フロッピーディスクの容量の都合で無理だろう。X680x0がCD-ROMに移行し損ねたというのは痛いといまさらながら思う。

また最新版の発表も思うに任せない。内部ではマイナーバージョンアップもしていたのだが，本体のアセンブラルーチンやSLASHをＣ言語から利用するためのライブラリ，それにライブラリのマニュアルを整備しているうちにまとまらなくなってしまい，まだ中途半端であるというのが実情。一応は動作するものの，きちんと公開できるレベルになっていない。というより，それをする元気がなくなってしまっているというのが大きい。というところで次に続く。

X680x0を振り返る

いつぞやの開発者インタビューで「CPUが速ければ専用ハードウェアは必要ない」という意味の主張があったことを記憶している。美しい理想である。しかしCPUが遅い部類に入ってしまっている現状ではまったく説得力がない。これはシャープというよりもモトローラの責任である。

考えてみるにモトローラの68000系を搭載したマシンはここ１年で潮が引くように勢いを失ってしまった。Amigaしかり，X680x0しかり。Macintoshはいいタイミングで PowerPCへ移行した。むろん68000系のMacintoshはフェーズアウトの段階にあるといっていいだろう。

なおMacintoshはOSのレベルでアプリケーションの作法を厳密に規定していたからPowerPCへの移行を比較的スムーズに行えたのであり，行儀の悪いプログラムが幅をきかせているX680x0に同じ手は通用しないだろう。もっとも，次期機種が出るとして，互換性を維持するためにDOSもどきのOSを載せ続けるような愚行をやってもらいたいとも思わない。

次期機種といえば，この連載は次期機種が出るまで引っ張ってそこで決着をつけようと考えていたのだ。しかし状況がこうなっては，現行機種で形にするしかない。はたして残り１カ月でなんとかなるのか，来月号に乞う御期待。

図４　レベル・オブ・ディテールを用いた表示

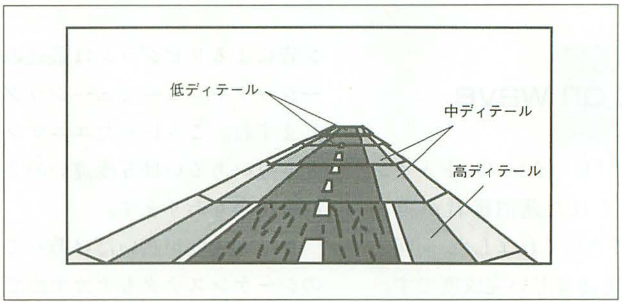


図５　周回数カウント処理の裏をかく

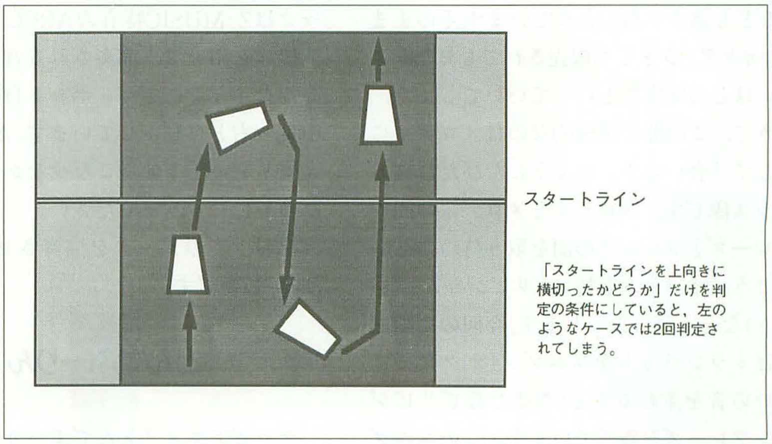
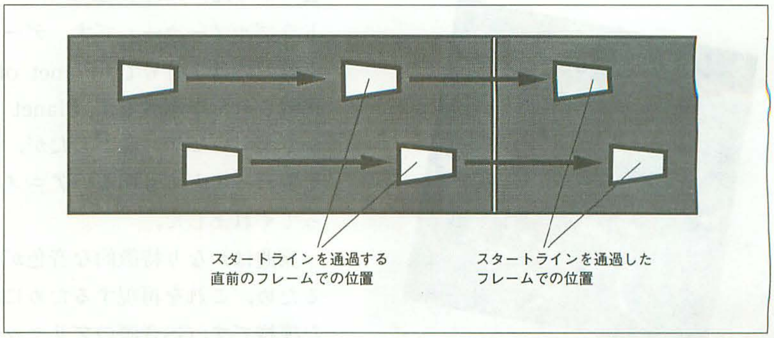


図６　高精度なラップタイム計測



Oh!X LIVE in '95

Z-MUSIC ver.2.0
SC-88対応

淋しい熱帯魚

Hasunuma Masaru 蓮沼 勝

Z-MUSIC ver.2.0
SC-55対応

「とんでぶーりん」より
ヒロインはトラブルメーカー

Kishimoto Hideaki 岸本 英昭

Z-MUSIC ver.2.0
SC-55対応

チャイコフスキー
弦楽のためのセレナーデII.Waltzer

Doi Atsushi 土井 淳史

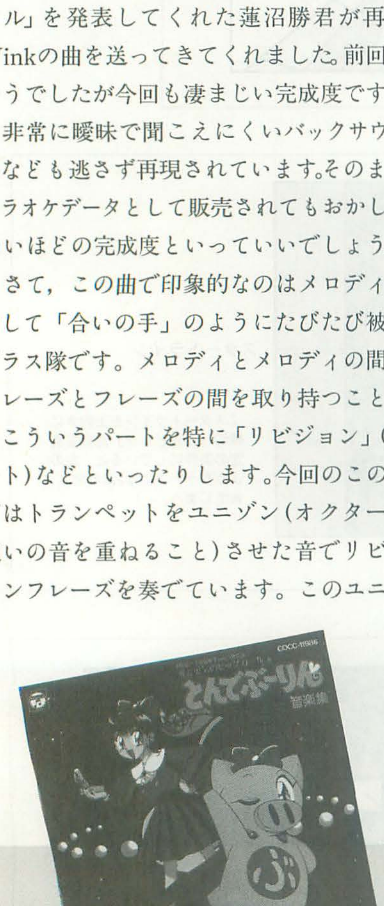
今月は蓮沼君のSC-88用データを以前に続きWinkで，テレビアニメより「とんでぶーりん」，そしてお馴染み土井君のチャイコフスキーです。今回は残念ながら内蔵音源の曲はなしということになってしまいました。

Heart on wave

6月号でWinkの「トゥインクルトゥインクル」を発表してくれた蓮沼勝君が再びWinkの曲を送ってきてくれました。前回もそうでしたが今回も凄まじい完成度です。

非常に曖昧で聞こえにくいバックサウンドなども逃さず再現されています。そのままカラオケデータとして販売されてもおかしくないほどの完成度といっていいでしょう。

さて，この曲で印象的なのはメロディに対して「合いの手」のようにたびたび被るブラス隊です。メロディとメロディの間やフレーズとフレーズの間を取り持つことからこういうパートを特に「リビジョン」(パート)などといったりします。今回のこの曲ではトランペットをユニゾン(オクターブ違いの音を重ねること)させた音でリビジョンフレーズを奏でています。このユニゾン音によるリビジョンは最近のダンスミュージックやニューミュージックでよく耳にしますね。こういったユニゾン音のほかに3度違いあるいは5度違いの2和音が使われる場合もあります。



とんでぶーりん音楽集

さて完成度が高いだけあってそのデータのシーケンステクもナカナカです。たとえばトラック8を見てみましょう。このトラックはZ-MUSIC特有のARCCと波形メモリ機能を用いて，演奏される音符とは独立したタイミングでパンポット(音場)を左←中央→右と移動させています。ARCC，波形メモリを上手く使いこなせなかった人はぜひ参考にしてみてください。

なお，このデータを演奏させるにはSC-88が必要です。

とんでぶーりん

テレビアニメ「とんでぶーりん」の主人公「かりんちゃん」のテーマ「ヒロインはトラブルメーカー」です。データを送ってくれたのは7月号で「Planet of life」で初掲載された岸本君です。Planet of lifeは静かで美しいピアノ曲でしたが，今回は打って変わってとても明るいアニメソングを送ってくれました。

原曲はかなり特徴的な音色が使われているため，これを再現するために相当苦労した模様です。GS音源のプリセット音だけでなんとかこれを再現すべく，いろいろな音の組み合わせを試してみた結果，現在の状態に落ち着いたとか。なるほどメロディは単音構成なのに「@91 Poly Synthe」と「@73 Piccolo」のユニゾン3和音で演奏されています。でも聞いてみるとユニゾンサウンドというよりはなにか別の音色の単音演奏のように聞こえます。単一音色間のユニゾンは，先ほどの「淋しい熱帯魚」のブラスリビジョンのようなギラギラした響きになるのですが，このような違う音色間のユニゾンはなにか別の新しい音色に聞こえることがあります。実はこれに近い音色の作り方を採用したシンセもあったりします。KORGのWaveStationがそうです。どーしてもほしい音色が見つからないとき，このような「音の調合」を試みるのもいいかもしれません。

演奏にはGS音源が必要です。なお，編集部でSC-55/SC-55mkII/SC-88による正常な演奏を確認しました。

チャイゲン

さて，最後もクラシック，チャイコフスキー「弦楽合奏のためのセレナーデ II.Waltzer」をお届けします。

1995年2月号でチャイコフスキー「白鳥の湖」を掲載，そして'95年8月号の付録ディスクには同じくチャイコフスキーの「ロミオとジュリエット」「組曲『白鳥の湖』」全15曲のZMSデータが収録された，Oh!X-

LIVEクラシックの申し子土井淳史君の投稿です。

クラシックはちょっと大きめのレコード屋の楽譜コーナーに行けばかなり正確な楽譜を手に入れることができます。しかし，彼のデータは単に楽譜をデータ化したものではないことはその演奏を聞けば容易にわかります。どこが普通のベタ打ちと違うのか見てみましょう。

1)　各楽器のパンポットで実際のオーケストラのポジションが再現されている

これは細かい点ですが，彼のデータはちゃんとバイオリンが左から，ビオラやコントラバスが右から鳴っています。聞き手がオーケストラに向かって座っている場合の楽器の配置をちゃんと再現しています。

2)　あたかも指揮者がいるように曲調に応じてテンポが変化する

これは案外DTM上級者でも忘れがちです。しかし，これをリアルに再現するにはよほど原曲を聞き込んでいるか，奏者の経験がないと難しいですよね。

3)　情緒豊かに各楽器の音量(ベロシティ)が刻々と変化する

クラシック音楽の醍醐味ともいえるダイナミックレンジの広さがあますことなく再現されています。か細い演奏から大迫力の演奏まで曲調に応じてリアルタイムに変化します。2)の効果とあいまって実にリアルな演奏になっています。

皆さんもたまには気分を変えてクラシック音楽に挑戦してみてはいかがですか。奥が深いですよ。

さて，演奏にはGS音源が必要です。編集部でSC-55/SC-55mkII/SC-88での正常な演奏を確認いたしました。　(Z.N.)

リスト1　淋しい熱帯魚

```
==================  HEART_ON_WAVE.ZMS  ==================
/ heart_on_wave.zms
/
/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/
/      淋しい熱帯魚  ／  W i n k
/
/      作詞：及川眠子 ／ 作曲：尾関昌也 ／ 編曲：船山基紀
/
/
/                          programmed by M.Hasunuma  1994/07
/
/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

.comment 淋しい熱帯魚 ／ W i n k  for SC-88 by M.Hasunuma

(i)(b1)

(m1,6000)(amidi10,1)
(m2,6000)(amidi10,2)
(m3,6000)(amidi10,3)
(m4,6000)(amidi14,4)
(m5,6000)(amidi2,5)
(m6,6000)(amidi4,6)
(m7,6000)(amidi1,7)
(m8,6000)(amidi3,8)
(m9,6000)(amidi3,9)
(m10,6000)(amidi7,10)
(m11,6000)(amidi5,11)
(m12,6000)(amidi6,12)
(m13,6000)(amidi8,13)
(m14,6000)(amidi9,14)
(m15,6000)(amidi11,15)
(m16,6000)(amidi12,16)
(m17,6000)(amidi13,17)

/====================================================

.roland_exclusive $10,$42 {0,0,127,0}
                   / SC-88 MODE 1

.sc55_reverb {  0  / reverb macro
          2        / reverb character
          4        / reverb pre-lpf
        120        / reverb level
         75        / reverb time
          0        / reverb delay feedback
         20}       / reverb send level to chorus

.sc55_chorus {  0} / chorus macro

.roland_exclusive $10,$42 {$40,$01,$51
             0     / delay pre-lpf
           116     / delay time center
            24     / delay time ratio left
            12     / delay time ratio right
            30     / delay level center
           100     / delay level left
           120     / delay level right
           100     / delay level
            80     / delay feedback
             0}    / delay sendlevel to reverb

.roland_exclusive $10,$42 {$40,$02,$00
             0     / eq low freq
            65     / eq low gain
             1     / eq high freq
            66}    / eq high gain

/====================================================
.wave_form 8,1 {  0,  8, 16, 24, 32, 40, 48, 56, 60,
                 52, 40, 28, 16,  4, -8,-20,-32,-44,
                -56,-60,-56,-48,-40,-32,-24,-16, -8}

.wave_form 9,0 {  0, 10, 20, 30, 40, 50,
                 60, 70, 80, 90,100}

(t1)/---------------------------------    drum
    @i$41,$10,$42t125
    n10i0,2@17@v100@u113p3q3@r1r
    @y$1c,50,16@y$1c,43,110@y$1a,37,127

    r1r1r2 o2L4 cccc cccc
    cccc cc'd8c'd8'd16c'd8.
    cccc ccL16'cd'ddr'cd'rdd

    |:L4
    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|c{'cd''cd'rc}
    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|{'cd''cd''cd'r 'cd'r'cd'c}2

    |:3c'cd':|c'cd'
    |:3c'cd':|{crcc}{'dc'ccc}
    |:3c'cd':|c'cd'
    |:3c'cd':|{cr'cd''cd'}{'cd''cd'cc}

    |:3c'cd':|c{'cd'u-15'cd''cd''cd'u}
    |:3c'cd':|c{'cd''cd'r'cd'}
    |:3c'cd':|c{'cd'u-15'cd''cd''cd'u}
    |:3c'cd':|{'cd''cd'rr}{'cd'r'cd''cd'}

    |:3c'cd':|c{'cd'rcc}
    |:3c'cd':|c{'cd'ccc}

    |:3c'cd':|{cr'cd''cd'}{'cd''cd'r'cd'} |
    |:3c'cd':|c{'cd'ccc}
    |:3c'cd':|{crr<d>}{crgr}

    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|c{'cd'ccc}

    c'cd'c{'cd'rcc}c'cd'
    {'cd''cd'cr}{'cd'r'cd''cd'} :|

    |:3c'cd':|c{'cd'ccc}
    c'cd'c{'cd'c}rrr{'cd'r'cd''cd'}

    |:3c'cd':|c{'cd'rcc}
    |:3c'cd':|c{'cd''cd'rc}
    |:3c'cd':|c{'cd'rcc}
    c'cd'c'cd'c'c8d'u127L16c+c+
    |:u127c+*0u113'cd':|
    u127 c+c+ u127c+*0u113'cd'r
    |:u127c+*0u113'cd':| u113L4

    |:3c'cd':|c{'cd'u-15'cd''cd''cd'u}
    |:3c'cd':|c{'cd''cd'r'cd'}
    |:3c'cd':|c{'cd'u-15'cd''cd''cd'u}
    |:3c'cd':|
    {'cd'u-10'cd''cd''cd'u}{'cd'r'cd''cd'}

    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|c{'cd'ccc}

    c'cd'c{'cd'rcc}'cd''cd'
    |:{'cd'u-15'cd''cd''cd'u}:|

    |:3c'cd':|c{'cd'rcc}
    |:3c'cd':|c{'cd'ccc}

    |:3c'cd':|{cr'cd''cd'}{'cd''cd'r'cd'}

    |:3c'cd':|c{'cd'ccc}
    |:3c'cd':|{crr'cd'}{'cd'r'cd''cd'}

    |:3c'cd':|c{'cd'rcc}
    |:3c'cd':|c{'cd'ccc}

    |:3c'cd':|{cr'cd''cd'}{'cd''cd'r'cd'}
    |:3c'cd':|c{'cd'ccc}
    c'cd'c{'cd''cd'}rrr{'dc'rcc}

    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|c{'cd''cd'rc}
    |:3c'cd':|c{'cd'r'cd'c}
    |:3c'cd':|{'cd''cd''cd'r 'cd'r'cd''cd'}2

```

```
    {'cd''cd'}ccc {'cd''cd''cd''cd'}2r2
(t2)/--------------------------------    hihat
    @i$41,$10,$42
    [k.sign +c,+f,+a]
    n10@u85p3q3@r1r
    r1r1
    @y$1c,46,10 @y$1c,42,10r2
    o2<c1>r1
    L16o4araaar8. r4rro1
    |:3'a!<c':|r|:'a!<c':| o2r2r<c8.>
    r1L16fffa a8ff fffa afff
    o2
    |:
    |:4 ffa8ffff ffa8ffff ffa8ffa8 ffffffaa :|
    |:8 ffa8frfr fffrfrfr ffa8frfr ffrfara8 :|
    |
    |:8 ffa8frfr ffa8frfr ffa8ffa8 fffffra8 :|
    :|
    |:5 ffa8frfr ffa8frfr |
    ffa8ffa8 fffffra8 :| r1
    |:4 ffa8frfr ffa8frfr ffa8ffa8 fffffra8 :|
    |:4 ffa8frfr fffrarfr ffa8frfr ffrfara8 :|
    |:13 ffa8frfr ffa8frfr |
    ffa8ffa8 fffffra8 :| r1
    |:4 ffa8ffff ffa8ffff ffa8ffa8 ffffffaa :|
    L8aar2.aafar2
(t3)/--------------------------------    maracas
    @i$41,$10,$42
    n10@u90p3q3@r1k1r
    r1@y$1c,70,95r1r2
    r1r1 r1r1 r1r1
    o4L16 |:
    |:60 aaaa aaaa :| |
    |:20 aaaa aaaa :| :|
    |:4  aaaa aaaa :|
    |:   aaaa aaa|a:|r r1
    |:76 aaaa aaaa :|
    |:4  aaaa aaaa :|
    |:   aaaa aaa|a:|r r1
    |:16 aaaa aaaa :|
    L8aar2.aaaar2
(t4)/--------------------------------    c.cym.
    @i$41,$10,$42
    n14@17x$40,$1d,$14,1,1@v100u85q2r
    r1r1r2
    r1r1 r1r1 r1r1
    |: o3L4
    p3c+rrr r1 r1r1 p1c+rrr r1 r1r1
    p2c+rrr r1 r1r1 r1r1 r1r1
    |:rp1c+rp2c+ u-30rc+urr p1c+rrr r1:|
    p1c+rrr r1 r1r1 r1r1 r1r1
    |r1r1 p2c+rrr r1 r1r1 r1r1 :|
    r2.r8p1c+8r2.p2c+4 c+rrr r1 r1r1
    p1c+rrr r1r1 r2p2c+4p1c+4
    |:rp1c+rp2c+ u-30rc+urr p1c+rrr r1:|
    |:p2c+rrr r1 r1r1 r1r1:|
    r1r1 r1r1
    p1c+rrr r1 r1r1 r1r1
    r1r1 r2.p1c+8p2c+8r2.c+4
    p3c+rrr r1 r1r1 p2c+rrr r1 r1r1
    p3L8c+c+r2.p1c+p2c+p1c+p3c+r2
(t5)/--------------------------------    bass
    @i$41,$10,$42
    n2i2,2@40@v100@u100p3@e5y126,1r
    r1r1r2
    o1L2|:aefg:| r1i9@39r1
    |:|: L8o1|:4a<a>:||:4g<g>:|
    |:4f<f>:|d<d>d<d>e<e>e<e>:|
    |:4a<a>:||:4g<g>:| |:4f<f>:||:4g<g>:|
    |:4a<a>:||:4g<g>:| f<f>f<f>g<g>g<g>|:4a<a>:|
    |:|:4f<f>:||:4g<g>:|
    |:4f<f>:|||:4e<e>:|:|g<g>g<g>e<e>e<e>
    |:8a<a>:||:8g<g>:|
    |:4f<f>:|g<g>g<g>e<e>e<e>
    |:4a<a>:||:4g<g>:| |
    d<d>d<d>e<e>e<e>|:4a<a>:|
    |:d<d>e<e>f<f>e<e>:|d<d>e<e>f<f>g<g>|:4a<a>:|
    |:8a<a>:| :|
    d<d>d<d>e<e>e<@y1,$64,84f&f2.r4@y1,$64,64>
    |:|:4a<a>:||:4g<g>:|||:4f<f>:||:4g<g>:|:|
    d<d>d<d>e<e>e<e>|:4a<a>:|
    |:|:4f<f>:||:4g<g>:||:4f<f>:||:4e<e>:|:|
    |:d<d>e<e>f<f>e<e>:|d<d>e<e>f<f>g<g>|:4a<a>:|
    |:8a<a>:|
    |:8a<a>:||:8g<g>:||:4f<f>:|g<g>g<g>e<e>e<e>
    |:4a<a>:||:4g<g>:|d<d>d<d>e<e>e<e>|:4a<a>:|
    |:8a<a>:||:8g<g>:||:4f<f>:|g<g>g<g>e<e>e<e>
    |:4a<a>:||:4g<g>:|
    d<d>d<d>e<e>e<@y1,$64,84f&f2.r4@y1,$64,64>
    |:|:4a<a>:||:4g<g>:|
    |:4f<f>:|d<d>d<d>e<e>e<e>:|
    aar2.aaaar2
(t6)/--------------------------------    hit / sax
    @i$41,$10,$42
    n4i0,2@56@v100@u110p3q6@e60,20r
    r1r1r2
    r1r1 r1r1 r1r1
    |:o4L8arrr r2 |:r2.r16aa16&
    a16r2...r1 |r1:| |:26r1:| |
    r1r1 r1r1 r1r1 :|
    |:28r1:|
    r1r1r1
    i126,1@56@v80u105@y1,$21,76@y1,$20,61
    r4o5L16@b-683,0'd*8>f+'&@b0'd*16>f+'
    @y1,$21,84@y1,$20,64
    u100@q3'd>f+'q5u105'c>e'u110
    |:q8@b-683,0'd*6>f+'&@b0'd*18>f+'q6'c>e'r:|
    @y1,$21,64
    r1r1 r1r1 r1r1
    r1r1 r1i0,2@56@v100u110q6r1
    o4L8rrrr r2 |:r2.r16aa16&
    a16r2...r1 |r1:|
    aar2.aagar2
(t7)/--------------------------------    kb
    @i$41,$10,$42
    n1i1,2@82@v100@u100@e60,15
    @y1,$63,65r16@y1,$66,68r16
    @y1,$20,68r16@y1,$21,72r16 p3
    o4L16q6'a>b''g>b''g>b'r'g8.>b'r
    'a8.c'r'a8c'r'a8c''g>b''g>b'r'g8.>b'
    |:4r'ac':|'a8c''g>b''g>b'r'g4>b'
    |:'a8.c'r'a8c'r'a8c''g>b''g>b'r'g4>b'
    |:4'ac'|r:|'a8c''g>b''g>b'r'g4>b':|
    r1r1 i25,2@57u90
    @y1,$63,65@y1,$66,82
    @y1,$20,80@y1,$21,75@e70,20
    |: |:
    o6L8q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'q8{'c>c''d>d''c>c'}8.
    q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'r8.
    o6L8q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'q8{>'a>a''b>b'<'c>c'}8.
    q3'f>f''g>g''e>e''c>c'
    q8'd8.>d'q3'e>e'r8. :| |:8r1:|
    o6|:rrq6'e>a'rrr'e>a'rr1rr'e>a'rr2|r1:|
    r2L16q6rr'e>b''e>b''d>a''e>b''gd'r
    L16o6q6'ae'rrrr2
    rr'c>c''d>d''e>e'r'e>e''d>d'r'd>d''c>c'r
    >'b>b'r'b>b'r'a>a'rrr
    r2.rr'b>b'<'c>c''d>d'r'd>d''c>c'r'c>c'>'b>b'r
    'a>a'r'a>a'r'g>g'rrr
    r1r1
    o6r2rr'e>e''d>d'r'd>d''e>e'r
    r2rr'd>d''c>c'r'c>c''d>d'r |
    r1q8|:rr'a>a'r:|r2 |:6r1:| :| |:6r1:|
    o5q5|:rr'ae'<'e>a'r'e>a'>'ae'rr2:|
    r1rr'ae'<'e>a'r'e>a'>'ae'rr2
    o6L8|:rrq6'e>a'rrr'e>a'rr1rr'e>a'rr2r1:|
    |:6r1:|
    L16o6r2.
    rr'c>c''d>d''e>e'r'e>e''d>d'r'd>d''c>c'r
    >'b>b'r'b>b'r'a>a'rrr
    r2.rr'b>b'<'c>c''d>d'r'd>d''c>c'r'c>c'>'b>b'r
    'a>a'r'a>a'r'g>g'rrr
    r1r1
    o6r2rr'e>e''d>d'r'd>d''e>e'r
    r2rr'd>d''c>c'r'c>c''d>d'r
    r1q8|:|:rr'a>a'r:|u-12:|u
    L16o6r2.
    rr'c>c''d>d''e>e'r'e>e''d>d'r'd>d''c>c'r
    >'b>b'r'b>b'r'a>a'rrr
    r2.rr'b>b'<'c>c''d>d'r'd>d''c>c'r'c>c'>'b>b'r
    'a>a'r'a>a'r'g>g'rrr
    r1r1
    o6r2rr'e>e''d>d'r'd>d''e>e'r
    r2rr'd>d''c>c'r'c>c''d>d'r
    r1r1
    |: o6L8q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'q8{'c>c''d>d''c>c'}8.
    q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'r8.
    o6L8q3'a>a''e>e''d>d''e>e'
    q8'd8.>d'q3'c>c'q8{>'a>a''b>b'<'c>c'}8.
    q3'f>f''g>g''e>e''c>c'
    q8'd8.>d'q3'e>e'r8. :|
    o6'a>a''a>a'r2.'a>a''a>a''g>g''a>a'r2
(t8)/--------------------------------    arp.1
    @i$41,$10,$42
    n3i16,2@7@v100@u100p3@e70r
```

▶「SX-WINDOW ver.3.1開発キット」を購入しました。購入目的は約200本におよぶSX-WINDOW用フリーソフト。ひと口に200といっても1つひとつ見ていくと，いい加減嫌になる量でした。とても贅沢なことですね。

多田　哲也(24)兵庫県

```
   m,1@c10,64,64s,8h,1@s,4@a1
   r1r1r2@q2
   o4L16|:4ar<erdrer>ar<edrder>:|
   @ap3
   o3i8@6r4|:6a<e>|<a>aru-10:|<a>

   |:i16@7u90

   @a1o4L16|:8ar<erdrer>ar<edrder>:|

   u80
   |:24ar<erdrer>ar<edrder>:| |
   |:ar<erdrer>ar<edrder>:|

   @ap3r1r1
   i0@47r1@a1o6 c>aec aec>a< ec>ae< c>aec
   @ap3r1r1 :|

   o4ar<erdrer>ar<edrder>r1

   |:16ar<erdrer>ar<edrder>:|

   @ap3r1r1
   i0@47r1@a1o6 c>aec aec>a< ec>ae< c>aec
   @ap3r1r1

   i16@7u80@a1o4L16
   |:19ar<erdrer>ar<edrder>:| r1

   |:9ar<erdrer>ar<edrder>:|

   ararrrar@ap3r2

(t9)/--------------------------------   arp.2
   @i$41,$10,$42
   n3@u120p3r
   m,1@c7,100,100s,9@s,2@a-1
   r1r1r2@q2
   o5L16|:4ar<erdrer>ar<edrder>:|
   @a@v100r1r1

   |: @a-1u110|:8ar<erdrer>ar<edrder>:|

   u100
   |:24ar<erdrer>ar<edrder>:| |
   |:ar<erdrer>ar<edrder>:|

   @a@v100r1r1r1o6ec>ae <c>aec aec>a< ec>ae
   r1r1 :|

   o5ar<erdrer>ar<edrder>r1

   |:16ar<erdrer>ar<edrder>:|

   @a@v100r1r1r1o6ec>ae <c>aec aec>a< ec>ae
   r1r1

   @a-1u100o5
   |:19ar<erdrer>ar<edrder>:| r1

   |:9ar<erdrer>ar<edrder>:|

   ararrrar@a@v100r2

(t10)/--------------------------------   e.tom
   @i$41,$10,$42
   n7i0,2@119@v100@u100p3@e10,0r
   r1r1r2 r1r1
   r1r2o5L8p1g@p48c>p3b16g8.
   r1i9@118r2L16o5ccc8c8cc
   |: |:6r1:| r1r2ccc8c8cc
   i3@128r4o6c8r8r2r1 |:22r1:|
   |r1i9@118r2o5{rrr@p16grr@p110cr}2p3
   r1r1 r1r1 r1r2 {ggcrgrgg}2 :|
   i9@118r1r2.p3{grgg}4
   r1r1 r1r2.{ggrc}4 r1r1 r1r2{ggrrgrgc}2
   r1r1 r1r1 r1r1 r1r2{gggggrgg}2
   r1r1 r1r1 r1r2 {gggggggg}2
   r1r1 r1r1 r1r1 r1r1 r1r2{rrrggrgg}2
   r1r1 r1r1 r1r1 r1r1 r2.{gc}4r2.g4
   r1r1 r1r1 r1r1 r1r2 ccc8c8cc
   L8ggr2.gec>gr2

(t11)/--------------------------------   synth
   @i$41,$10,$42
   n5i0,2@103@v100@u65p3@e60@q2r
   r1r1r2

   r1r1r1
   r2...o6'a1ed>a',4>'a1ed>a'r16

   |: r1r1
   r1i0@101u74r1
   o5|:'a2.e>a',0<<'e4>e'>>'a1e>a':|

   r1r1r1r1 r1r1r1r1

   L4o5|:r'<ae>ae':|r4.L8egab<c>
   a4'<a4e>ae'r2r4.'e<e'b<c>bg
   L4'e<e'|:'<ae>ae'r:|L8r'e<e'gab<c>
   L4'e<e''<ae>ae'r2r1

   r1r1r1r1 r1r1 r1r1r1r1|

   r1r1r1r2...'a1<ea<e',4'a1e>ae'r16 :|

   o5L8u70'a2.e',0u80<'e>e'>'b>b'&
   'b2.>b'<'e>e'>'a>a'&
   'a2.>a'<'e>e''e>e'&'e4>e''d4>d'>'b4>b''g4>g'

   'e2.>e''a>a'<'e>e'&'e2.>e'
   'd>d'>'a4.>a'<'c>c'>'b4.>b'<'e>e'>'a>a'&'a1>a'

   u74
   L4o5|:r'<ae>ae',0:|r4.L8egab<c>
   a4'<a4e>ae'r2r4.'e<e'b<c>bg
   L4'e<e'|:'<ae>ae'r:|L8r'e<e'gab<c>
   L4'e1<e'r1

   o5r1r1r1r2...'a1<ea<e',4'a1e>ae'r16

   r1r1r1r1 r1r1 r1r1r1r1
   r1r1r1r1 r1r1 r1r1r1r1

   i0@101u74
   o5|:4'a2.e>a',0<<'e4>e'>>'a1e>a':|

   r1r1

(t12)/--------------------------------   se
   @i$41,$10,$42
   n6x$40,$16,$14,1,2
   i0,2@58@v100@u120p3r
   r1r1r2

   r1r1r1r1
   r1@y1,$63,90@y$18,50,56
   o3d2
   @y1,$63,64@y$18,50,64@y$18,86,38
   'd2d+o6d'

   |: |:8r1:|

   @51@y$18,67,20@y$18,66,60@p50r4o4'g2.f+'r1
   |:24r1:| | r1r1 r1r1 r1r1 :|

   |:16r1:|

   i0@58@u100p3
   r1r1r1r1
   r1@y1,$63,90@y$18,50,56 o3d2
   @y1,$63,64@y$18,50,64@y$18,86,38
   'd2d+o6d'

   |:16r1:|

   r1r1r1@y1,$63,90@y$18,50,56@y$18,51,50
   r4u120 'o3d2.d+'

   |:10r1:|

(t13)/--------------------------------   clav.
   @i$41,$10,$42
   n8i0,2@8@v100@u75p2@e60r
   @y1,$21,90@y1,$66,74
   r1r1r2 r1r1r1r1 r1r1

   |:
   o2L16 |:
   ar<ar>ar<ar> ar<aa>g<grg>
   gr<gr>gr<gr> g<gg>g<gg>g<g>
   fr<fr>fr<fr> fr<ff>f<ff>f
   d<ddrdrdd> e<eereree> :|

   o3L16
   r1r2rrggr>g<gg r1r2rrggr>g<gg
   r1r2rraar>g<gg r1r2>aa<aa>aa<aa

   r1r2rrgg>gg<gg r1r2rrgg>gg<gg
   r1r2rrgg>rg<gg r1r2rree>re<er

   |:r2rraa>ra<aa:||:r2rrgg>rg<gg:|

   r2rreer>b<ee r2rrerddre |

   r2rragrgar r2rrgfrfgr
   r2rreer>e<ee r2rrerr>e<ee

   r2rrffreee r2rrffreee
   r2rrffrggg r2raargaaa

   r1r1 :|

   r2rragrgaa r2rrgfrfgg r1 r1

   r2rraar>a<aa r2rrggr>g<gg
   r2rrffr>f<ff r2rrggr>g<gg
   r2rraar>a<aa r2rrggr>g<gg
   r2rrffr>f<ff r2rrarr>aar<

   r1r2rrgg>rg<gg r1r2rree>re<ee
   r1r2rrgg>rg<gg r1r1

   r2rrffreee r2rrffreee
   r2rrffrggg r2raaraara

   r1r1

   |:r2rraa>ra<aa:| |:r2rrgg>rg<gg:|

   r2rreer>b<ee r2reereere

   r2rragrgar r2rrgfrfgr
   r2rreer>e<ee r2rrerr>e<ee

   |:r2rraa>ra<aa:| |:r2rrgg>rg<gg:|

   r2rreer>b<ee r2reereere

   r2rragrgaa r2rrgfrfgg r1r1
```

▶ NewXに望むのは「美しさ」です。外観でも，OSでも，付属ソフトでもいいから，「美しさ」を見せてください。 松尾 繁(21)福岡県

```

    o2L16|:
    ar<ar>ar<ar> ar<aa>g<grg>
    gr<gr>gr<gr> g<gg>g<gg>g<g>
    fr<fr>fr<fr> fr<ff>f<ff>f
    d<ddrdrdd> e<eereree> :|

    r1r1

(t14)/--------------------------------   strings
    @i$41,$10,$42
    n9i0,2@49@v100@u96p3@e60r
    r1r1r2 r1r1 r1r1 r1r1

    |: |:
    o4L8|:q3aede q8d8.q3cq8|{cdc}8.:|r8.
    q3aede q8d8.q3cq8{>ab<c}8.
    q3fgec q8d8.q3er8. :|

    |:16r1:|

    o4q6L16
    r2.rrcd eredrdcr >brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr |
    r1q8rrarrrarr2

    r1r1 r1r1 r1r1 :| r1r1

    o4u50c1>b1a1b1<
    c1>b1a2b2<c1

    |:14r1:|

    o4q6L16
    r2.rrcd eredrdcr >brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr
    r1q8|:4rrar:|

    o4q6L16
    r2.rrcd eredrdcr >brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr
    r1r1

    |:
    o4L8|:q3aede q8d8.q3cq8|{cdc}8.:|r8.
    q3aede q8d8.q3cq8{>ab<c}8.
    q3fgec q8d8.q3er8.
    :|

    aar2.aagar2

(t15)/--------------------------------   brass
    @i$41,$10,$42
    n11i0,2@62@v100@u80p1@e65r
    r1r1r2 r1r1 r1r1 r1r1

    |:
    p1u80 |:
    o4L8|:q3aede a|dr16q8{cdc}8.:|r16q3er8.
    aede aeda frfr grgr :|

    |:16r1:|

    o4q6L16
    r2.rrcd eredrdcr >brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr |
    r1q8rrarrrarr2

    r1r1 r1r1 r1r1 :|

    r1r1 r1r1 r1r1
    o4q5|:rr'ae'<'e>a'r'e>a'>'ae'rr2:|
    r1rr'ae'<'e>a'r'e>a'>'ae'rr2

    |:14r1:|

    o4q6L16
    r2.rrcd eredrdcr> brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr
    r1q8rrar rrarr2

    o4q6L16
    r2.rrcd éredrdcr> brbrarrr
    r2.rrb<c drdcrc>br arargrrr

    r1r1

    r2rragrgar r2rrgfrfgr
    r1r1

    |:
    o4L8|:q3aede a|dr16q8{cdc}8.:|r16q3er8.
    aede aeda frfr grgr
    :|

    aar2.aagar2

(t16)/--------------------------------   vocal
    @i$41,$10,$42
    n12i8,2@6@v100@u98p3@e95,20r@g12@j1
    @y1,$64,82@y1,$66,70
    @y1,$63,60@y1,$21,75@e,10y126,1@m20
    m,1@c7,100,100s,3
    r1r1r2

    r1r1 r1r1 r1r1

    r1r1 r1r1
    r1r1r1u98r2.L8ro5q4y94,30d
    @b-1366,0e16&@b0q8e.y94,0r4rq4eed4&ccd&ed4
    >b&<c4q8c>q3aq8<cd4d&e4.f16&e16&d4rq4y94,30d
    @b-1366e16&@b-1366,0e16&@b0q8e
    y94,0r4rq4eed4&g4g&aq8g4r
    q7d4q5dq4eq8d&@b0,-2732d>@b0ga&b<c16&>b16&a4r2

    u108o5rrq4aq6grrq4aq6g
    rq4aq5be4q8g4&@b0,-2049g24&@b-2049g24r24
    r@b0rq7aq5gq7aq8brq6e4@b-2049,0g16&
    @b0g8.@b-1366,0e32&@b0e16.&q8d4r4
    rrq4aq6grrq4aq6g
    rq4aq5@b-1366,0b32&@b0b16.e4q8g&
    @b0,-2049g&@b-2049g16r16
    r@b0rq7aq5gq7aq8br@b-1366,0b16&
    @b0b4^16&_30beq8~e4r4

    L8q8a@b-1366,0a32&@b0a8..
    @b-1366,0a32&@b0a16.&q7a2r1
    q8gg4@b-1366,0a32&@b0a8..q7g4.r1

    q8@b-683,0,12f&@b0fq4fcfg4
    @b-1366,0,0g16&@b0g16&q6g2a-2

    q4gaaq8@b-1366,0a16&@b0a4^16&@b0,-683ar@b0
    q4gq6g4@b-1366,0a16&@b0a.q8g4r

    @b-683,0f16&@b0f16&eq4fagq8g4
    @b-1366,0a16&@b0a16&a4r4
    o5q6a4<q8c4

    |:q8@b-1366,0d16&@b0d16dd16&@b0,-683d16q5@b0c
    q8d4u-18@b-1366,0e32&@b0e16.ru:|
    @b-1366d&@b-1366,0d32&@b0d16.dcd&cc4&c2.r8>

    |:4u-30q3guq8@b-1366,0a16&@b0a16_20
    |1p2:||2p1:||3p2:||4~80p3:|
    |:4u-30q3guq8@b-1366,0a16&@b0a16_20
    |1p2:||2p1:||3p2:||4r:|


    r1@v100p3r1r1r1
    r1r1r1u98r1L8o5q4y94,30
    @b-1366,0e16&@b0q8e.y94,0r4rq4eed4&ccd&ed4.
    q7c4q8c>q3aq8<cd4d&e4.f16&e16&d4rq4y94,30d
    @b-1366e16&@b-1366,0e16&@b0q8e
    y94,0r4rq4eeq6d4g4g&aq8g4r
    q7dq4ddfq8e4>gb4<c16&>b16&q5a2r4

    u108o5rrq4aq6grrq4aq6g rq4aq5be4q8g4.
    rrq7aq5gq7aq8brq6e4@b-2049,0g16&@b0g8.
    @b-1366,0e32&@b0e16.&q8d4r4
    rrq4aq6grrq4aq6g rq4aq5
    @b-1366,0b32&@b0b16.e4q8g4.
    rrq7aq5gq7a@b-1366,0b32&@b0b8..q4b
    q8b4.u-20eq8u+20e4r4

    L8q8a@b-1366,0a32&@b0a8..
    @b-1366,0a32&@b0a16.&q7a2r1
    q8gg4@b-1366,0a32&@b0a8..q7g4.r1

    q8@b-683,0,12f&@b0fq4fcfg4
    @b-1366,0,0g16&@b0g16&q6g2a-2

    q4gaaq8@b-1366,0a16&@b0a4^16&@b0,-683ar@b0
    q4gq6g4@b-1366,0a16&@b0a.q8g4r

    @b-683,0f16&@b0f16q4ffaq6g4gq8
    @b-1366,0a16&@b0a16&a1&
    @s,30@a100a2r2 @a@v100r1 r1r1 r1r1 r1r1

    u108o5rrq4aq6grrq4aq6g rq4aq5be4q8g4.
    rrq7aq5gq7aq8brq6e4@b-2049,0g16&@b0g8.
    @b-1366,0e32&@b0e16.&q8d4r4
    rrq4aq6grrq4aq6g rq4aq5
    @b-1366,0b32&@b0b16.e4q8g4.
    rrq5aq5gq7abre4g4r
    o5q6a4<q8c4

    |:q8@b-1366,0d16&@b0d16dd16&@b0,-683d16q5@b0c
    q8d4u-18@b-1366,0e32&@b0e16.ru:|
    @b-1366d&@b-1366,0d32&@b0d16.dcd&cc4&c2.r8>

    |:4u-30q3guq8@b-1366,0a16&@b0a16_20
    |1p2:||2p1:||3p2:||4~80p3:|
    |:4u-30q3guq8@b-1366,0a16&@b0a16_20
    |1p2:||2p1:||3p2:||4r:|

    @v100p3o5L8q8a@b-1366,0a32&@b0a8^16.
    @b-1366,0a16&@b0a16&q7a2r1
    q8gg4@b-1366,0a16&@b0a8.q7g4.r1

```

▶「AI-68K」……以前からなんだろうこれは？　と思っていました。なるほど，言語だったんですね。でも，Lispもいいけど C＋＋入門もお願いします。

佐井川　泰治(23)東京都

```
    q8@b-683,0,12f&@b0fq4fcfg4
    @b-1366,0,0g16&@b0g16&q6g2a-2

    q4gaaq8@b-1366,0a16&@b0a4..r@b0
    q4gq6g4@b-1366,0a16&@b0a.q8g4r

    @b-683,0f16&@b0f16&fq4fagq8g4
    @b-1366,0a16&@b0a16&
    a4&@b0,-341a16&@b-341a16rr2@b0

    o5L8q8a@b-1366,0a32&@b0a8..
    @b-1366,0a32&@b0a16.&q7a2r1
    q8gg4@b-1366,0a32&@b0a8..q7g4.r1

    q8@b-683,0,12f&@b0fq4fcfg4
    @b-1366,0,0g16&@b0g16&q6g2a-2

    q4gaaq8@b-1366,0a16&@b0a4..r@b0
    q4gq6g4@b-1366,0a16&@b0a.q8g4r

    @b-683,0f16&@b0f16q6fq4faq8g4q6gq8
    @b-1366,0a16&@b0a16&a1&
    @s,30@a100a2r2 @a@v100r1 r1r1 r1r1 r1r1

    r1r1

(t17)/--------------------------------   chorus
    @i$41,$10,$42
    n13i2,2@82@v100@u80p3@e90,20r@g12@j1
    m,1@c7,100,100s,3
    r1r1r2

    r1r1 r1r1 r1r1

    |:16r1:|

    u80o5L8q4|:|:rr'e>a''e>g':|r1|r1r4.q8
    @b-1366e32&@b0e16.d4r4:|r1r2e4r4

    o5L8q7'e>a'@b-1366,0'e32>a'&
    @b0'e..>a''e>a'&'e2>a'r1
    'd>g''d4>g'@b-1366,0'e16>a'&
    @b0'e8.>a''d4.>g'r1

    q8c4q4c>a<cq7d4q8@b-1366e16&
    @b-1366,0e16&q6@b0e2d2

    q4deeq8@b-1366,0e&@b0e4.r
    q4dq6d4@b-1366,0e&@b0eq8d4r
    q6d4q4dedq8d4c&c4r4 u70o4q6'a4c''a4c'

    |:q8@b-1366,0'd16<d'&@b0'd16<d''d<d''d16<d'&
    @b0,-683'd16<d'q5@b0'c<c'
    q8@b-683,0'd16<d'&@b0'd.<d'@b-1366,0'e16<e'&
    @b0'e16<e'r:|
    @b-1366'd<d'&@b-1366,0'd32<d'&
    @b0'd16.<d''d<d''c<c'
    'd16.<d'&@b0,-1366'd32<d'&@b-1366'd<d'
    @b0'c4<c'&'c2.<c'r4

    r1r1


    |:16r1:|

    u80o5L8q4|:|:rr'e>a''e>g':|r1|r1r4.q8
    @b-1366e32&@b0e16.d4r4:|r1r2e4r4

    o5L8q7'e>a'@b-1366,0'e32>a'&
    @b0'e..>a''e>a'&'e2>a'r1
    'd>g''d4>g'@b-1366,0'e16>a'&
    @b0'e8.>a''d4.>g'r1

    q8c4q4c>a<cq7d4q8@b-1366e16&
    @b-1366,0e16&q6@b0e2d2

    q4deeq8@b-1366,0e&@b0e4.r
    q4dq6d4@b-1366,0e&@b0eq8d4r
    q6dddeq8d4q4dq8c&c1&
    @s,20@a100c2r2 @a@v100r1r1r1
    r1r1r1r1

    u80o5L8q4|:|:rr'e>a''e>g':|r1|r1r4.q8
    @b-1366e32&@b0e16.d4r4:|r1r2
    u70o4q6'a4c''a4c'

    |:q8@b-1366,0'd16<d'&@b0'd16<d''d<d''d16<d'&
    @b0,-683'd16<d'q5@b0'c<c'
    q8@b-683,0'd16<d'&@b0'd.<d'@b-1366,0'e16<e'&
    @b0'e16<e'r:|
    @b-1366'd<d'&@b-1366,0'd32<d'&
    @b0'd16.<d''d<d''c<c'
    'd16.<d'&@b0,-1366'd32<d'&@b-1366'd<d'
    @b0'c4<c'&'c2.<c'r4
    r1r1

    o5L8q7'e>a'@b-1366,0'e32>a'&
    @b0'e..>a''e>a'&'e2>a'r1
    'd>g''d4>g'@b-1366,0'e16>a'&
    @b0'e8.>a''d4.>g'r1

    q8c4q4c>a<cq7d4q8@b-1366e16&
    @b-1366,0e16&q6@b0e2d2

    q4deeq8@b-1366,0e&@b0e4.r
    q4dq6d4@b-1366,0e&@b0eq8d4r
    q6d4q4dedq8d4c&c2r2

    o5L8q7'e>a'@b-1366,0'e32>a'&
    @b0'e..>a''e>a'&'e2>a'r1
    'd>g''d4>g'@b-1366,0'e16>a'&
    @b0'e8.>a''d4.>g'r1

    q8c4q4c>a<cq7d4q8@b-1366e16&
    @b-1366,0e16&q6@b0e2d2

    q4deeq8@b-1366,0e&@b0e4.r
    q4dq6d4@b-1366,0e&@b0eq8d4r
    q6d4q4deq8d4q6dq8c&c1&
    @s,20@a100c2r2 @a@v100r1 r1r1 r1r1 r1r1

    r1r1


(p)
```

リスト2　淋しい熱帯魚カウンタ表示

```
 1:00006510 00000000   2:00006510 00000000   3:00006510 00000000   4:00006510 00000000
 5:00006510 00000000   6:00006510 00000000   7:00006510 00000000   8:00006510 00000000
 9:00006510 00000000  10:00006510 00000000  11:00006510 00000000  12:00006510 00000000
13:00006510 00000000  14:00006510 00000000  15:00006510 00000000  16:00006510 00000000
17:00006510 00000000
```

リスト3　とんでぶーりん

日本音楽著作権協会(出)許諾第9571707-501号

```
===============  HE_TBMK.ZMS  ===================
.comment とんでぶーりん『ヒロインはトラブルメーカー』for GS
(i)
.sc55_init
.sc55_reverb $10={1,4,0,90,72,64,64}

(m1,1000)(aMIDI1,1)
(m2,1000)(aMIDI1,2)
(m3,1000)(aMIDI2,3)
(m4,1000)(aMIDI3,4)
(m5,1000)(aMIDI3,5)
(m6,1000)(aMIDI4,6)
(m7,1000)(aMIDI5,7)
(m8,1000)(aMIDI5,8)
(m9,1000)(aMIDI6,9)
(m10,1000)(aMIDI7,10)
(m11,1000)(aMIDI8,11)
(m12,1000)(aMIDI9,12)
(m13,1000)(aMIDI11,13)
(m14,1000)(aMIDI11,14)
(m15,1000)(aMIDI12,15)
(m16,1000)(aMIDI10,16)
(m17,1000)(aMIDI10,17)
(m18,1000)(aMIDI10,18)
(m19,1000)(aMIDI10,19)
(t1,2,3,4,5,6,7,8)  t144@i$41,$10,$42
(t9,10,11,12,13,14) t144@i$41,$10,$42
(t15,16,17,18,19)   t144@i$41,$10,$42

/----- シンセ 1 -----
(t1) o4@u60i0,1@91@v120
(t2) o5@u80i0,1@91@v120
(t3) o6@u80i0,0@73@v95
(t1,2,3)
      y91,70y93,30@p64 r4|: L8q4 ra<c&>arfa&fr
      df&dq8eefgq4 ra<c&>arfa&f rq8L16dred|c8frfrr4:|
      crfrfrr4 |:12r1:|_10L8
/----- シンセ 2 -----
(t4) @u90i0,1@89 @v120
(t5) @u80i0,1@89 @v120
(t6) @u90i0,0@101@v95
(t4,5,6)
      y91,70y93,30@p64q8 r4 |:8r1:|o5
(t4,6)
      L8fra-rgrb-r e-rge-ffr4 q4d-e-fa-g4q8<d->g frge-r2
      fra-rgrb-r e-rb-ga-a-r4 r<c>a-fa-r<c>b-
```

▶ Amigaユーザーの友人が「X68000がほしい」といいだしました。私が洗脳したんです。
でも，わたしはX68000をもっていない。　　　　川手　隆義(21)神奈川県

```
(t4) o5@u60@v115
(t5) o6@u80@v115|:7r1:|
(t6) o6@u80@v90
(t4,5,6)
      |:8r1:|L8 r8q8L16dredcrfrfrr4
/----- シンセ3 -----
(t7) o4@u92i0,1@49@v120
(t8) o5@u92i0,1@49@v120
(t9) o5@u92i0,1@50@v95
(t7,8,9)
      y91,70y93,30@p50q8 r4 |:16r1:| L4
      c2d2& db-ed8e8 c2a2& ab-<dc> _5L8
(t1,2,3,7,8,9)
      q4 ra<c&>arfa&fr df&dq8eefgq4 ra<c&>arfa&f r2ffr4
/----- ベース1 -----
(t10) y91,50y93,10@p40o2@u99i8,0@40y99,1y98,32y6,70L16@v105
       r4|:fffr8.frd8d8r4 >b-b-b-r8.b-r<c8c8r4
           fffr8.frd8d8r4 >b-b-b-r<c8.r>f8f8r4:| |:8r1:|L16
        <|:fffr8.frd8d8r4 >b-b-b-r8.b-r<c8c8r4fffr8.frd8d8r4
         |>gggr8.gr<q8c8c8r4:| >b-b-b-r<c8.r>f8f8r4
/----- ベース2 -----
(t11) i0,0@34@v100 y99,1y98,32y6,70
(t12) i0,0@37@v115 y99,1y98,32y6,44
(t11,12)
      y91,50y93,10@p40o2q8 |:8r1:| L16r4
      @u120d-8<d-r8.@u110d-r>  @u120e-8<e-r8.@u110e-r>
      @u120c 8<c r8.@u110c r>> @u120f 8<f r8.@u110f r>
      @u120b-8<b-r8.@u110b-r   @u120e-8<e-r8.@u110e-r>>
      @u120g 8<g r8.>@u110g r  @u120a-8<a-r8.>@u110a-r<
      @u120d-8<d-r8.>@u110d-r  @u120e-8<e-r8.>@u110e-r
      @u120c 8<c r8.>@u110c r> @u120f 8<f r8.>@u110f r
      |:4@u120b-8<@u110b-r:| @u120|:4q8e-8<@q3e-e-:||:8r1:|
/----- ピアノ -----
(t13) y91,70y93,30@p100o4i0,0@2y99,1y98,100y6,54L2@v95@q12r4
      @u100'cfa''dfa' 'dfb-''c4eg'@u110'>g4<ce'@u100
      >'a<cf''a<df' @u110'f4b-<d''g4<ce'q4'a<cf'@q12<
      @u100'cfa''dfa' 'dfb-''e4g<c'@u110'c4eg'@u100
      'cfa'>'a<df' @u110'f4b-<d''g4<ce'q4'fa<c'@q12
      @u95'fa-<d-''gb-<e-' 'gb-<e-''fa-<c'
      'b-<d-f''b-<e-g' 'b<dgb'<'ce-a-<c'
      @u100>'a-4<d-f<d-''<d-4fa-''gb-<e-'
      'g4b-<e-''b-4<e-g''a-<cf' 'b-<cd-f''fb-<cd-'
      @u80'gb-<e-<''gb-<e-<'@u100
      |:'a<cf''a<df' 'b-<df''g4<ce''<c4eg' 'a<cf''a<df'|
      'gb-<df''g<ce':|'f4b-<d''g4<ce'@u110q4'a8<cf''a8<cf'r4
/----- ピアノ(一部) -----
(t14) r4 |:15r1:|
      @u125o5L4@q12'd-<d-''e-<e-''f<f''g<g' |:8r1:|
/----- ギター -----
(t15) y91,50y93,0@p20o3i16,1@29@v105@q2@u110L16r4 |:
      r8aa<crfr>r8aa<drfr> r8b-b-<drfrr8ccergr>
      r8aa<crfr>r8aa<drfr> r8b-b-<crer>ara*28r*44:|
      r8a-a-<d-rfr>r8b-b-<e-rgr> r8b-b-<e-rgrr8ccfra-r>
      r8b-b-<d-rfr>r8b-b-<d-rgr> r8bb<drfrr8cce-ra-r>
      r8a-a-<d-rfrr8>b-b-<e-rgr> r8b-b-<e-rgrr8ccfra-r>
      r8b-b-<d-rfrr8d-d-fra-r> r8ggb-r<e-rr8>b-b-<e-rgr>
      |:r8aa<crfr>r8aa<drfr> r8b-b-<drfr>|r8b-b-<crer>:|
      <r8ccergr> r8aa<crfr>r8aa<drfr> r8b-b-<drfrr8ccergr>
      r8aa<crfr>r8aa<drfr> r8b-b-<crer>ararr4
/----- ドラム -----
(t16) y91,120y93,30@v120L4r4
      |:@u80o3a*0@u100o2 |:3cdc8c8d:|
      |cdc8c8d8d16d16:| cdd8d8@u90o3c16c16o2a8L8
      |:@u80o3a*0@u100o2|:3ccd4ccd4:|
      |ccd4ccd*0@u90ca@u100:| ccd4cL16ddd8ddL4
      |:@u80o3a*0@u100o2|:3cdc8c8d:||cdc8c8d:|cd@u110d8d4.
(t17) L16y99,24y98,77y6,62o5@u60r4
      r8f8r8fff8r4ff r4.ffr4fff8 r8f8r8ffr8f8r8ffr4.ffr4fff8
      r8f8r8fff8r4ff r4.ffr4.ff  r8f8r8ffr8f8r8ffr4.ffr4.ff
      |:6r4.ff:| r4ffffr4.ff |:3r4.ff:|r4ffff |:4r4.ff:|
      |:11r4.ff:|r4ffff |:r4.ff:| r8f8r8ffr2
(t18) L8@u63o2[K.SIGN +f,+a]r4 y99,28y98,42y6,44 y98,46y6,44
      fffffaff fffffffa ffafffaf ffaffaff
      ffffffaf fffffffa ffafffaf ffaffaff
      fffffffa fffffaff fffffffa ffffffaf
      fffffaff fffffffa fffffffa fffffffa
      fffffffa fffffaff fffffffa fffffffa
      ffafffaf ffafffaa fffaffa ffffff4.
(t19) @u40o3[K.SIGN +f]r4 |:8r1:| L16
      |:8|:3f8ff:|ffff :| |:4r1:|
      |:7f8ff:|ffff |:3f8ff:|ffff |:f8ff:|f8f4.
(p)
```

リスト4 とんでぶーりんカウンタ表示

```
 1:00001230 00000000  2:00001230 00000000  3:00001230 00000000  4:00001230 00000000
 5:00001230 00000000  6:00001230 00000000  7:00001230 00000000  8:00001230 00000000
 9:00001230 00000000 10:00001230 00000000 11:00001230 00000000 12:00001230 00000000
13:00001230 00000000 14:00001230 00000000 15:00001230 00000000 16:00001230 00000000
17:00001230 00000000 18:00001230 00000000 19:00001230 00000000
```

リスト5 Waltzer

```
================ TGEN2.ZMS ===================
.comment [SC55 only] チャイコフスキー 弦楽のためのセレナー
デ II.Walzer
.comment P.Tschaikowsky "SERENADE for String Orchestra" Mo
derate Tempo di Valse
.comment       arranged by Doi Atsushi, For SC-55 Apr 1994
(i)
(d0)
/caigencaigencaigencaigencaigencaigencaigencaigencaigencaig
/
/   S E R E N A D E
/            f o r  S t r i n g  O r c h e s t r a
/
/   I I . W a l t e r
/            Moderato Tempo di Valse
/                        P.Tschaikowsky
/
/        チャイコフスキー  s=============mm=
/                          ──────────────
/  弦楽のためのセレナーデ                ノ￣"--ˉ "i
/                               ≒ ÷==≡ ≡ ≡==|=⇔:
/                                      ヽ__w⌒,_α
/
/       第2楽章:ワルツ
/
/            arranged by Doi Atsushi
/             製作:土井 淳史
/      F o r  S C - 5 5
/
/       1994  4/14(Thu) ~ 1994  4/15(Fri)
/
/TCHAIKOTCHAIKOTCHAIKOTCHAIKOTCHAIKOTCHAIKOTCHAIKOTCHAIKOTC
/
/-----------------------------------------------------------
(b1)                / Base Channel = MIDI
/-----------------------------------------------------------
/       Track Map
/ Strings
/  1st Violin      Ch:1    Tr:1,2
/  2nd Violin      Ch:2    Tr:3,4
/  Viola           Ch:3    Tr:5,6
/  Violoncello     Ch:4    Tr:7,8
/  Doublebass      Ch:5    Tr:9,10
(m 1,4000)(aMIDI1 , 1)
(m 2,2000)(aMIDI1 , 2)
(m 3,4000)(aMIDI2 , 3)
(m 4,2000)(aMIDI2 , 4)
(m 5,4000)(aMIDI3 , 5)
(m 6,2000)(aMIDI3 , 6)
(m 7,2000)(aMIDI4 , 7)
(m 8,2000)(aMIDI4 , 8)
(m 9,2000)(aMIDI5 , 9)
/-----------------------------------------------------------
/ SC55 INIT
.roland_exclusive 16,66={$40,00,$7F,00}
/-----------------------------------------------------------
/VOICE RESERVE
.sc55_v_reserve $10={6,6,4,4, 4,0,0,0 ,0,0,0,0 ,0,0,0,0}
/       REVERB SET
.sc55_reverb $10={4,4,0,65,90,0,0}
/       CHORUS SET Macro Pre-lpf Level Feedback Delay Rate De
pth Send level to Chorus
.sc55_chorus $10={2,0,100,20,80,3,19,0}
/-----------------------------------------------------------
/ MML DATA SET
(t1)t 200 @i$41,$10,$45
/*** 1st Violin
(t1)       i0@ 49@p 32@v100y91,70y93,80@j1k  0@C11,127,127 @
h,0 @s,1
(t1)@y1,99,64,0                                  r1
(t2)                 @v100                 @j1k  0r1
/*** 2nd Violin
(t3) r16    i0@ 49@p 48@v 90y91,70y93,80@j1k  0@C11,127,127 @
h,0 @s,1
(t3)@y1,99,55,0                                  r2r4..
```


▶新連載の「Lisp一夜漬け」が面白い。できれば，文法の解説だけで終わらず，「Lispではこんなことができる！」というところまでやってくれたらと思う。

加藤 潔(26)大阪府

```
(t4) r16                @v 90              @j1k  0r2r4..
/*** Viola
(t5) r8    i0@ 49@p 72@v 90y91,70y93,80@j1k  0@C11,127,127 @
h,0 @s,1
(t5)@y1,99,55,0                                 r2r4.
(t6) r8                 @v 90              @j1k  0r2r4.
/*** Violoncello
(t7) r8.   i0@ 49@p 88@v 95y91,70y93,80@j1k  0@C11,127,127 @
h,0 @s,1
(t7)@y1,99,55,0                                 r2r4r16
(t8) r8.                @v 95              @j1k  0r2r4r16
/*** Double bass
(t9) r4    i0@ 49@p112@v 95y91,70y93,80@j1k-12@C11,127,127 @
h,0 @s,1
(t9)@y1,99,55,0                                 r2r4
(t 1)[!]
(t 2)[!]
(t 3)[!]
(t 4)[!]
(t 5)[!]
(t 6)[!]
(t 7)[!]
(t 8)[!]
(t 9)[!]
/
/
/  T e m p o   d i   V a l s e
/
/
/ Violin I
/II:Walzer
(t1)X$10,$00,$00,$49,$49,$3a,$57,$61,$6c,$7a,$65,$72
(t1)t 160
(t1)r4@u64o3q8L4b<c t 190@u+4d@ue@uf+ @ug2@ua
(t1)@ub8r8L16@A127d&@A122d&@A116d&@A110d&@A104d&@A98d&@A92d&
@A86d
(t1)L4r@Ay11,127@u64ga b8{<c>b}8ab <d8r8>gb a2.
(t1)r@u+4ab <c8{dc}8>b<c @ue8r8>a<c >b8r8<d2& @udc+>b a8{ba}
8g+a @ub8r8c+b
(t1)@ua8r8<d2& @ud8r8c+>b @ua8{ba}8g+ t-10 a @u96<t 130g2t 1
70c+
(t2)|:19r2.:|
/ Violin II
(t3)r2. r4@u48o4q4L4dd rc+c+ r2. rdd rdd r2g rf+f+ rf+g
(t3)r@u+4gg r2a r@ugg re+e+ r@uf+f+ r@uee r@uaa r@ugg r@uf+f
+ r@u80ee
(t4)r2. r4@u48o3q4L4bb r2. rbb rbb rbb |:3r2.:|
(t4)r@u+4gg <rdd |:3r2.:| r@u64>aa <r@u+4dd r2.r2. >r@u80gg
/ Viola
(t5)r2. @u48o3q4L4|:5rgg:| <rdd |:q7>d<q4dd:|
(t5)r@u+4cc rcc r@udd rdd r@udd r2@uc+ >r@uaa <r@udd r@udd r
@u80c+c+
(t6)r2. @u48o3q4L4rdd ree |:3rdd:| rgg raa rab r2.r2.
(t6)r@u56bb rbb |:3r2.:| r@u72bb r@u+4aa r2.
/ Violoncello
(t7)r2. @u48o2q7L4|:5gr2:| b2r r@u72q8L8<<c2& c@u-8ec>aq4gf+
(t7)@u48o3q8L2e. @u+4f+. g. @ug+. @ua. @ug. @uf+. @ug. @ua.
>@u80a.
(t8)|:19r2.:|
/ Double Bass
(t9)r2. @u48o2q7L4|:5gr2:| b2r <dr2 dr2
(t9)@u+4q8L2e. f+. @ug. g+. @ua. @ug. @uf+. @ug. @ua. >@u80a
.
/
/   [ A ]
/                                    ( 20/223 9.0% )
/
/ Violin I
(t1)t 190
(t1)@u96o5q8L8d2.& dq4@y1,99,55@u112c+def+g @u+4q8a2@u-8@y1,
99,64f+4 @u+8a2@u-8f+4
(t1)@u+8{b@u-8ag}4a2& a4@u112f+4.g a2a4& a4f+4.g a2a4& a4f+4
.g t-10 a2a4& t-10 a4a2 t-10 a2a4&
(t1)t-10 a4 t 40 L16@A127a&@A120a&@A112a&@A104a&@A96a&@A88q6
a t 120@Ay11,127@u64L8q8g
(t1)t 190 L8f+4g4a4 {g&a&}gf+4e4 drg2& g4f+4d4 crf+2& f+4e4c
4
(t1)>b@y1,99,55<@u+4q4c@ud@ue@ud@ue
(t1)@uf+@ug@u96 t60 L16q8@A127a&@A120a&@A112a&@A104a&@A96a&@
A88q6a
(t1)t 120@Ay11,127@y1,99,64@u64L8q8g
(t1)t 190 f+4g4a4 {g&a&}gf+4e4 dr@u+4g2& g4f+4e4 @u72drf+2&
f+4c+4e4 dr@u+4f+2& f+4c+4e4
(t1)@u80dc+>bf+d@u+4f+ bf+b<@uc+dc+ df+@ubf+b<c+ @y1,99,55@u
96L16|:6d&e&:|
(t2)|:34r2.:|
/ Violin II
(t3)@u80o4q4L4rdd L8ar@y1,99,55r@u112o5q4c+de @u+4q8f+2@y1,9
9,64@u-8d4
(t3)@u+8f+2@u-8d4 {@u+8g@u-8f+e}4f+2& f+4@u112d4.e f+2f+4& f
+4d4.e f+2f+4& f+4d4.e
(t3)f+2f+4& f+4f+2& f+2f+4&
(t3)f+4 L16@A127f+&@A120f+&@A112f+&@A104f+&@A96f+&@A88q6f+ @
Ay11,127@u64L8q8e
(t3)d4e4f+4 {e&f+&}ed4c4 >br<e2& e4d4>b4 ar<d2&
(t3)d4c4>a4 g@y1,99,55q4a@u+4b<@uc>@ub<@uc
(t3)@ud@ue @u112q8L16@A127f+&@A120f+&@A112f+&@A104f+&@A96f+&
@A88q6f+ @Ay11,127@y1,99,64@u64L8q8e
(t3)d4e4f+4 {e&f+&}ed4c4 >br<@u+4e2& e4d4c+4 >br<@u72d2&
(t3)d4>a+4<c+4 >br<@u+4d2& d4>a+4<c+4 @u80>br4.r4 r2.r2. r4@
u104o3q8L8b4<c4
(t4)@u80o3q4L4raa <f+8r4.r4 |:32r2.:|
/ Viola
(t5)r4@u80o3q4L4f+<f+ d8r4.r4 |:12r2.:|
(t5)r8@u48o3q4L8d<c>ada <cc+@u+8q7d2 r@u-8q4>dbgdg b<c@u+8q7
d2
(t5)r@u-8q4>d<c>ada <cc+@u+8q7d2 @u-8dr4.r4 r2.
(t5)rq4>d<c>ada <cc+@u+8q7d2 r@u-4>q4dbgdg @u+4b<c+q8d4e4 @u
f+q4d>bf+b<d
(t5)@uq8f+2.& f+q4@ud>bf+b<d @uq8f+2.& f+r4.r4 |:3r2.:|
(t6)|:20r2.:| @u48o3q7L8gr4.r4 |:13r2.:|
/ Violoncello
(t7)@u80o3q8L8d2.& dr4.r4 |:10r2.:| @y1,99,64 r2.r2.
(t7)@46@u40o3q8L4|:adr:| |:gdr:| f+dr adr br2 r2.
(t7)|:adr:| @u+4gdr g>gr @u48<f+>f+r
(t7)@y1,99,55r8@49@u64o2q4L8f+<f+c+>a+f+ <f+r>f+r4. r@u+8f+<
f+c+>a+f+
(t7)@u80br4.r4 r2.r2. @y1,99,64r4@u104o2q8L4b<c
(t8)|:20r2.:| @u40o3q8L4dr2 |:13r2.:|
/ Double Bass
(t9)@u80o3q8L8d2.& dr4.r4 |:10r2.:| @y1,99,64 r2.r2.
(t9)@46@u40o3q8L4|:adr:| |:gdr:| f+dr adr gr2 r2.
(t9)|:adr:| @u+4gdr g>gr @u48<f+>f+r
(t9)@y1,99,55r8@49@u64o2q4L8f+<f+c+>a+f+ <f+r>f+r4. r@u+8f+<
f+c+>a+f+ @u80br4.r4 |:3r2.:|
/
/   [ B ]
/                                    ( 54/223 24.2% )
/
/ Violin I
(t1)t 190
(t1)@u96o6q8L16|:4d&e&:|{d&e&d&e&d}4 L8c+4.>c+cq4<c >bagd>bg
(t1)q7dr4.r4 <q8L16|:10g&a&:|{g&a&g&a&g}4 L8f+q4ed>af+e q7dr
4.r4 r2.
(t1)q5L4rdd r2. L8r>a+b<c+de+ f+c+def+a <c+>ea<c+de f+c+ref+
a ba+b<c+de f+c+def+g
(t1)t-20 a t-10 a+bage t 190 d>af+d>af+
(t2)|:9r2.:| @u96o4q5L4rcc r>bg |:8r2.:|
/ Violin II
(t3)@u104o4q8L8d4e4f+4 g2a4 brd2 r4g4a4 b&{<c&>b}a4b4 <dr>g4
b4 a2.
(t3)r4a4b4 <c&{d&c}>b4<c4 er>a4<c4 >br<d2& d4c+4>b4 a&{b&a}g
+4a4 brc+4b4
(t3)ar<d2& d4c+4>b4 a&{b&a}g+4a4 <g2c+4 d2.&
(t4)|:19r2.:|
/ Viola
(t5)@u96o4q5L4rdd ree- r2. |:5rdd:| rgg L8o3rdef+ga bf+gab<c
+
(t5)drL4e+e+ rf+d ree r2. rgg rf+f+ ree rf+f+
(t6)@u96o3q5L4rbb rb-a |:rbb:| |:rgg:| |:rf+f+:| rgg r2.r2.
rg+g+ r2. |:raa:|
(t6)rgg <rdd >|:raa:|
/ Violoncello
(t7)@u104o3q8L8d4e4f+4 g2a4 brd2 r4g4a4 b&{<c&>b}a4b4 <dr>g4
b4 a2.
(t7)r4a4b4 <c&{d&c}>b4<c4 er>a4<c4 >br<d2& d4c+4>b4 a&{b&a}g
+4a4 brc+4b4
(t7)ar<d2& d4c+4>b4 a&{b&a}g+4a4 <g2c+4 d2.&
(t8)|:19r2.:|
/ Double Bass
(t9)@46@u80o2q8L4|:5g<g>r:| b<br d<dr> d<dr> @y1,99,55@49@u9
6o3q8L8e2. f+2. g2.
(t9)g+2. q7ar>ar4. <gr>gr4. <f+r>f+r4. <gr>gr4. |:<ar>ar4.:|
 <dr<q5d4d4
/
/   [ C ]
/                                    ( 73/223 32.7% )
/
/ Violin I
(t1)t 190
(t1)@u96o4q7L8dr@u80<q8{d&e&d}c+q4de q8f+dq4>bgf+d >br4.r4 r
2.
(t1)r4<<@u+8q8{c+&d&c+}>b+<q4c+a q8geq4c+>ged q8c+2. q7d+r4.
r4
(t1)r4<@u80q8{d+&e&d+}c++q4d+<c @u+4>q8bf+q4d+>bq7f+4 r4r4.q
4<@ub q8ae@uq4c+>aq7e4
(t1)r4r4r@u96<q4a q8gdq7c+4rq4g @u-4q8f+c+q7>b4r@uq4<f+ q8e>
b@ua+4b4
(t1)@u80q4<c+>e+f+a+<c+g q8f+dq4>bgf+d >q7br4.r4 r2.
(t1)r4<<{c+&d&c+}>b+<q4c+a @u+4q8geq4c+>ged @uq8c+2. @uq7d+r
4.r4
```

▶10月号の特集でサンプルのESPER MACH印刷を見て……本当に印刷か？
曹　之祐(21)東京都

```
(t1)r4<@u{d+&e&d+}c++q4d+<c @u>q8bf+q4d+>bq7f+4 r4<@uq8{e+&f+&e+}eq4e+<d
(t1)@uq8c+>g+q4e+c+>q7g+4 r4<@u112q8{f++&g+&f++}f+q4f++<e
(t1)q8d+>a+q4f++d+>q7a+4 r4r4.q4<<@u-6d+ q8c+>g+@uq4e+c+q7>g+4
(t1)r4r4.q4@u<<c+ q8>bf+q4@ud+>bq7f+4 r4r4.<@uq4b q8ae@uq4c+>aq7e4
(t1)r4r4.@u64q4<d q8agq4dc>q7g4
(t2)|:38r2.:|
/ Violin II
(t3)@u104o5q7L8dr4.r4 r2. @y1,99,55r2. r4@u84o4q8L4g2 @u-8f+@u+8e2& e2r4
(t3)r2. r4a2 @u-8g@uf+2 r4r4.@u84o5q4L8d+ q8f+eq4>b@u+4gq7e4 r4r4.<q4c+ q8ed@uq4>af+q7d4
(t3)r@u96q4bq8a+f+q7c+4 rq4@u-4a+q8ge>q7b4 r@uq4<gq8f+e@uq4d>b q7a+r4.r4 r2.r2.
(t3)r4@u84o4q8L4g2 @u-8f+@u+8e2& e2r r2. ra2
(t3)q4L4@u85gf+f+ @u+5f+f+r @uf+g+g+ @ug+g+r @u+6g+a+a+
(t3)r4r4.@u112o5q4L8f++ q8a+g+q4d+>bq7g+4 r4r4.<@u-4q4e+ q8g+f+@uq4c+>aq7f+4
(t3)r4r4.@uq4<d+ q8f+e@uq4>bg+q7e4 r4r4.q4@u<c+ q8edq4@uc>aq7d4 r4r4.<@u64q4g
(t4)|:26r2.:| @u95o3q4L4bbb @ubbb @u+6<c+c+c+ |:9r2.:|
/ Viola
(t5)@u96o4q7L4f+8r4.r4 @u72o3q4bbb rbb @u73|:12bbu+1:| @u93<c+>b<@u+1c+ d@uc+r
(t5)@u88>b<c+r >@u-8f+er @u-8g<c+r q7e8r4.r4 @u72>q4bbb r|:8b:|
(t5)bbb @u+4bbb @ubbb @ubbb @ubbb @u<d+c+c+ @uc+c+c+ @ue+d+d+
(t5)@u104d+d+d+ d+d+d+ c+c+c+ @u-8c+c+c+ @u>bbb @ubbb @uaaa @u<dr2 @u64dr2
(t6)@u96o3q7L4a8r4.r4 @u72o3q4|:6f+:| @u73|:6ggu+1:| |:4aau+1:|@u83a g@u+1gg r2.r2.
(t6)@u88eer r2. @u72c+f+f+ q7f+8r4r4.
(t6)@u72q4|:6f+:| |:6g:| ggg @u+4ggg @uaaa @uaaa @uaaa |:3r2.:|
(t6)@u104o4c+c+c+ >bbb bbb @u-8aaa @uaaa @ug+g+g+ @uggg @u<cr2 @u64>gr2
/ Violoncello
(t7)@u104o4q8L8dr4.@y1,99,55r4 @u80o3q4d4d4d4 q5dc+dc+d>b <q7f+r@u+8e4.@u-8q8d c+r4.r4 r2.
(t7)r@u88q5d+ed+ec+ q7gr@u+8f+4.@u-8q8e d+r4.r4
(t7)@u80o4q4L4d+c+d+ @u+4ed+e @u80>aaa @u+4aaa @u88gf+r @u-8a+br @ubc+>b q7f+8r4r4.
(t7)@u80o3q4d4d4d4 L8q5dc+dc+d>b <q7f+r@u+8e4.@u-8q8d c+r4.r4 r2.
(t7)r@u88q5d+ed+ec+ q7gr@u+8f+4.@u-8q8e d+r4.r4
(t7)r4q8{d+&e&d+}c++q4d+>b @u+6<q7a4g+2 r4@uq8{e+&f+&e+}eq4e+c+ @uq7b4a+2
(t7)@u112o4q4L4f++e+f++ g+f++g+ e+d+e+ @u-8f+e+f+ @ud+c+d+ @ued+e @uc+>b<c+
(t7)>@uar2 @u64<cr2
(t8)|:11r2.:| @u88o3q4L4ggg @u+4f+f+f+ r2. @u88ddr |:21r2.:|
(t8)@u72o3q4L4f+r2 @u64er2
/ Double Bass
(t9)@u96o4q7L8dr4.r4 r2.r@u80o3q5c+dc+d>b <q7f+r@u+8e4.@u-8q8d c+r4.r4 r2.
(t9)r@u88q5d+ed+ec+ q7gr@u+8f+4.@u-8q8e d+r4.r4
(t9)@u80o3q4L4aaa @u+4ggg @uggg @uf+f+f+ @u96eer @u-8ddr @uc+c+>b q7f+8r4.r4 r2.
(t9)@u80o3q5L8rc+dc+d>b <q7f+r@u+8e4.@u-8q8d c+r4.r4 r2.
(t9)r@u88q5d+ed+ec+ q7gr@u+8f+4.@u-8q8e d+r4.r4
(t9)r4q8d+c++q4d+>b @u+6<q7a4g+2 r4@uq8e+eq4e+c+ @uq7b4a+2
(t9)@u112o4q4L4c+c+c+ >bbb bbb @u-8aaa @uaaa @ug+g+g+ @uggg @uf+r2 @u64er2
/
/   [ D ]
/                                  ( 111/223 49.8% )
/
/ Violin I
(t1)t 190
(t1)r2@u64o5q8L8f+e q4c>br2 q8baq4f+er4
(t1)@y1,99,64r4@u64o4q8L4b<c @u+4d@ue@uf+ @ug2@ua
(t1)@ub8r8L16@A127d&@A122d&@A116d&@A110d&@A104d&@A98d&@A92d&@A86d
(t1)L4r@Ay11,127@u64ga @u+4b8{<c>b}8ab @u<d>gb @ua2.&
(t1)a@u80ab <c8{dc}8>b<c @u+2e8r8>a<c @u>b8r8<d2& @udc+>b @ua8{ba}8g+a @ub8r8c+b
(t1)@ua8r8<d2& @udc+>b @u96a8{ba}8g+ t-10 a <t 150g2t 150c+
(t2)|:22r2.:|
/ Violin II
(t3)@u64o5q8L8baq4f+dr4 r4q8dc>q4ag r2q8gf+
(t3)q4e-d@y1,99,64@u64o3q8L4b<c @u+4d@ue@uf+ @ug2@ua
(t3)@ub8r8L16@A127d&@A122d&@A116d&@A110d&@A104d&@A98d&@A92d&@A86d
(t3)L4r@Ay11,127@u64ga @u+4b8{<c>b}8ab @u<d8r8>gb @ua2.&
(t3)a@u80ab <c8{dc}8>b<c @u+2e8r8>a<c @u>b8r8<d2& @udc+>b @ua8{ba}8g+a @ub8r8c+b
(t3)@ua8r8<d2& @udc+>b @u96a8{ba}8g+a <g2c+
(t4)|:22r2.:|
/ Viola
(t5)@u64o4q4L4dr2 r2.r2.
(t5)r4@u64o4q8L8e-dq4>ba q8<c>ba+b<ed c>b<dc+gf+ ag<c>ba+b
(t5)f+g>bb-<c+c dc+gf+@u+2ag <c>b<@uedc+d @ugf+d>a@uf+a e+f+@ugf+c+d
(t5)@u80gcdd+eg @u+2c>b<cdef+ @uagf+g<c>b @ua+bfefd @ugf+e+f+c+d
(t5)@uagf+edc+ @udef+gg+a @ua+bfefd @u96gf+e+f+gf+ <dc+>bb-ag
(t6)@u64o4q4L4cr2 |:21r2.:|
/ Violoncello
(t7)@u64o3q4L4f+r2 @y1,99,64|:3r2.:|
(t7)@46@u48o2q8L4g<d>b g<ee- dgd >b<dg @u+2>g<d@ug d>@ubg <@udf+@ua <d>@uaf+
(t7)@u64eg<c @u+2>f+d<d @u>gb<d @u>g+b<d @u>>a<df+ @ugag @uf+af+ @ugg+<d
(t7)@u80>ad>a <ea>a
(t8)|:22r2.:|
/ Double Bass
(t9)@u64o3q4L4f+r2 r2. @y1,99,64r2.r2.
(t9)@46@u48o2q8L4g<d>b g<ee- dgd >b<dg @u+2>g<d@ug d>@ubg <@udf+@ua <d>@uaf+
(t9)@u64eg<c @u+2>f+d<d @u>gb<d @u>g+b<d @u>>a<df+ @ugag @uf+af+ @ugg+<d
(t9)@u80>ad>a <ea>a
/
/   [ E ]
/                                  ( 133/223 59.6% )
/
/ Violin I
(t1)t 190
(t1)@u96o6q8L8d2.& dq4@y1,99,55@u112>c+def+g @u+4q8a2@u-8@y1,99,64f+4 @u+8a2@u-8f+4
(t1)@u+8{b@u-8ag}4a2& a4@u112f+4.g a2a4& a4f+4.g a2a4& a4f+4.g t-10 a2a4& t-10 a4a2 t-10 a2a4&
(t1)t-10 a4 t 40 L16@A127a&@A120a&@A112a&@A104a&@A96a&@A88q6a t 120@Ay11,127@u64L8q8g
(t1)t 190 L8f+4g4a4 {g&a&}gf+4e4 drg2& g4f+4d4 crf+2& f+4e4c4
(t1)>b@y1,99,55<@u+4q4c@ud@ue@ud@ue
(t1)@uf+@ug@u96 t60 L16q8@A127a&@A120a&@A112a&@A104a&@A96a&@A88q6a
(t1)t 120@Ay11,127@y1,99,64@u64L8q8g
(t1)t 190 f+4g4a4 {g&a&}gf+4e4 dr@u+4g2& g4f+4e4 @u72drf+2&f+4c+4e4 dr@u+4f+2& f+4c+4e4
(t1)@u80dc+>bf+d@u+4f+ bf+b<@uc+dc+ df+@ubf+b<c+ @y1,99,55@u96L16|:6d&e&:|
(t1)|:4d&e&:|{d&e&d&e&d}4 L8c+4.>c+cq4<c >bagd>bg
(t1)q7dr4.r4 <q8L16|:10g&a&:|{g&a&g&a&g}4 L8f+q4ed>af+e q7dr4.r4 r2.
(t1)q4L4rdd r2. rfq8f q4f+f+f+ rf+f+
(t1)o5q5L8rf+g+a+b<c+ def+>f+f+<c+ dd+e>aab <c>bagf+e dc>bagb
(t1)<c+>b-gf+gc+ d>bg<dgb <c>ef+a<da
(t2)|:43r2.:| @u96o4q4L4rcc r>bg rg+q8g+ q4aaa ra+a+ |:8r2.:|
/ Violin II
(t3)@u96o5q8L8d2.& dr@y1,99,55r@u112o5q4c+de @u+4q8f+2@y1,99,64@u-8d4
(t3)@u+8f+2@u-8d4 {@u+8g@u-8f+e}4f+2& f+4@u112d4.e f+2f+4& f+4d4.e f+2f+4& f+4d4.e
(t3)f+2f+4& f+4f+2& f+2f+4&
(t3)f+4 L16@A127f+&@A120f+&@A112f+&@A104f+&@A96f+&@A88q6f+ @Ay11,127@u64L8q8e
(t3)d4e4f+4 {e&f+&}ed4c4 >br<e2& e4d4>b4 ar<d2&
(t3)d4c4>a4 g@y1,99,55q4a@u+4b<@uc>@ub<@uc
(t3)@ud@ue @u112q8L16@A127f+&@A120f+&@A112f+&@A104f+&@A96f+&@A88q6f+ @Ay11,127@y1,99,64@u64L8q8e
(t3)d4e4f+4 {e&f+&}ed4c4 >br<@u+4e2& e4d4c+4 >br<@u72d2&
(t3)d4>a+4<c+4 >br<@u+4d2& d4>a+4<c+4 @u80>br4.r4 r2.r2. r4@u104o3q8L8b4<c4
(t3)@u104o4q8L8d4e4f+4 g2a4 brd2 r4g4a4 b&{<c&>b}a4b4 <dr>g4b4 a2.
(t3)r4a4b4 <c&{d&c}>b4<c4 er>a4<c4 >br<d2& d4>b4<c+4 d&{e&d}c+4d4 f+rc+4e4
(t3)drf+2& f+4d4e4 f+&{g&f+}e4f+4 arc4d4 >br<g2& g4>g4a4 b&{<c&>b}a4b4 <e4d4>f+4
(t4)|:56r2.:|
/ Viola
(t5)@u96o4q8L8f+eq4d>af+a dr4.r4 |:12r2.:|
(t5)r8@u48o3q4L8d<c>ada <cc+@u+8q7d2 r@u-8q4>dbgdg b<c@u+8q7d2
(t5)r@u-8q4>d<c>ada <cc+@u+8q7d2 @u-8dr4.r4 r2.
(t5)rq4>d<c>ada <cc+@u+8q7d2 r@u-4>q4dbgdg @u+4b<c+q8d4e4 @uf+q4d>bf+b<d
(t5)@uq8f+2.& f+q4@ud>bf+b<d @uq8f+2.& f+r4.r4 |:3r2.:|
(t5)@u96o4q5L4rdd ree- r2. |:5rdd:| rgg L8o3rdef+ga bf+gab<d fedc+>bb-
(t5)a<df+a<df+ edc+>c+c+<c+ >brq4L4dd r2c+ rcc rdd rdd rc+c+ rdd rcc
(t6)|:20r2.:| @u48o3q7L8gr4.r4 |:13r2.:|
(t6)@u96o3q5L4rbb rb-a |:rbb:| |:rgg:| |:rf+f+:| rgg |:5r2.:|
(t6)rbb rf+f+ rdd rf+d rgg r2. rgg raa
/ Violoncello
(t7)@u80o3q4L4df+a <d8r4.r4 |:12r2.:|
(t7)@u40o3q8L4|:adr:| |:gdr:| f+dr adr br2 r2.
(t7)|:adr:| @u+4gdr g>gr @u48<f+>f+r
(t7)@y1,99,55r8@49@u64o2q4L8f+<f+c+>a+f+ <f+r>f+r4. r@u+8f+<
```

```
f+c+>a+f+
(t7)@u80br4.r4 r2.r2. @y1,99,64r4@u104o2q8L4b<c
(t7)@u104o3q8L8d4e4f+4 g2a4 brd2 r4g4a4 b&{<c&>b}a4b4 <dr>g4
b4 a2.
(t7)r4a4b4 <c&{d&c}>b4<c4 er>a4<c4 >br<d2& d4>b4<c+4 d&{e&d}
c+4d4 f+rc+4e4
(t7)drf+2& f+4d4e4 f+&{g&f+}e4f+4 arc4d4 >br<g2& g4>g4a4 b&{
<c&>b}a4b4 <e4d4>f+4
(t8)|:20r2.:| @u48o3q8L4dr2 |:35r2.:|

/ Double Bass

(t9)@u80o3q4L4df+a <d8r4.r4 |:12r2.:|
(t9)@u40o3q8L4|:adr:| |:gdr:| f+dr adr gr2 r2.
(t9)|:adr:| @u+4gdr g>gr @u48<f+>f+r
(t9)@y1,99,55r8@49@u64o2q4L8f+<f+c+>a+f+ <f+r>f+r4. r@u+8f+<
f+c+>a+f+
(t9)@u80br4.r4 r2. @y1,99,64r2.r2.
(t9)@46@u96o2q8L4|:5g<gr>:| b<br d<dr >d<dr @y1,99,55>@49q8e
2. f+2. g2. g+2.
(t9)q7ar2 a+r2 br2 b-r2 ar2 dr2 gr2 ere- dr2 dr2

/
/   [ F ]
/                                        ( 189/223 84.8% )
/

/ Violin I

(t1)[!]t 190
(t1)@u96o5q7L4gr2 |:4r2.:| @y1,99,64rq8f+g ab<c def+
(t1)g2. f2. e2. @u-8e-2. @ud2. @uc+2. @uc2. @u>b2a @u48gr2 r
2.r2.
(t1)r4L8q8{f+&g&f+}ef+a g4r2 |:3r2.:| @u40|:r4>{g&a&g}f+gb <
e4r2:|
(t1)r>c+dgbf+ gb<d>a+b<d gc+dgbf+ gb<d>a+b<d g4r2 @46o3q8L4'
g<g<db'r2 gr2
(t2)[!]|:35r2.:|

/ Violin II

(t3)[!]@u104o4q7L4gr2 r2.r2. @y1,99,64r@u96q8ga b<cd
(t3)e2. f+2. fee- L8@y1,99,55dq4c>bab<c q7dr4.r4 |:3r2.:|
(t3)@y1,99,64@u64o4q8L4rf+g ab<c @u-4d@ue@uf+ r2.
(t3)r4@u48o4q8L8{g&a&g}f+gb <e2.& L4edc >bq4dd ree >rgg
(t3)r4q8L8<{f+&g&f+}ef+a g4r2 @46@u32o4q8L4er2 >gr2 <er2 >gr
2
(t3)o6d>bg d>gg >g<gb >'b<g<d'r2 <'db<g'r2 r2.
(t4)[!]|:20r2.:| @u48o3q4L4rbb rbb r2.r2. q8gr2
(t4)@u32o3q8L4gr2 r2. gr2 r2. o5bgd >br2 r>g<d |:3r2.:|

/ Viola

(t5)[!]@u96o4q4L4dr2 @y1,99,64r@u96o3q8L4ab <cde f+2. f2. e2
. e-2.
(t5)dc+c @y1,99,55L8>bq4agf+ga q7br4.r4
(t5)r2. @y1,99,64r4@u80o3q8L4ga @u-4b<cd @ue2. @uf+2. @ufe@u
e-
(t5)@u48dq4dd ree ree rdd r2.
(t5)r4q8L8>{g&a&g}f+gb <e2.& e4d4c4 >b4r2 @u40|:r4{e&f+&e}d+
eg b4r2:|
(t5)@46@u32o5q8L4gd>b gbd dd>'g<g' 'g<db'r2 'db'r2 r2.
(t6)[!]@u96o3q4L4br2 |:15r2.:| r@u48o3q4L4bb rgg <rcc rcc |:
10r2.:|
(t6)r@u32o4q8L4d>b br2 r2.r2.r2.

/ Violoncello

(t7)[!]@u104o3q8L4g2. @u96f2. e2. e-2. d2. c+2. c2. >bb-a gr
2
(t7)rab @u-8<cde @uf+2. @uf2. @ue2. @ue-2. @udc+c
(t7)@u48q4>b<gg >e<bb caa daa ggg eee caa daa dr2
(t7)@46@u32o3q8br2 'db'r2 br2 'db'r2 r2.r2.
(t7)gb'db' >'g<db'r2 <gr2 >gr2
(t8)[!]|:18r2.:| @u48o3q4L4rgg rf+f+ r2.r2. ree rf+f+ |:11r2
.:|

/ Double Bass

(t9)[!]@u96o2q4L4|:9g<gg>:| @u90|:7g<gg>u-6:|
(t9)@u48gr2 er2 <cr2 dr2 gr2 er2 cr2 d<dd >gr2
(t9)@y1,99,64@46@u32o4q8L4c+r2 >gr2 <c+r2 >gr2 |:r2.:|
(t9)@u32o4q8L4rd>b gr2 gr2 >gr2

(t1)[!]t 60|:6r4:|

/
/1 2  3   4   5   6   7   8   9  10  11  12  13  14  15
/0 8 16  24  32  40  48  56  64  72  80  88  96 104 112 120
/                        pp          p  mp  mf       f      ff  fff
/timp                                       f        ff
/tri                 p              f      ff

/----------------------------------------------------------------
(p)
```

リスト6 Walzerカウンタ表示

```
1:00007F50 00000000   2:00007E30 00000000   3:00007E30 00000000   4:00007E30 00000000
5:00007E30 00000000   6:00007E30 00000000   7:00007E30 00000000   8:00007E30 00000000
9:00007E30 00000000
```

(善)の「勝負はこれからだ」

善バビ番外編

最初に，読者から同人CDが送られてきたので紹介しよう。

●MYU-TATION/VEL
CD:MYUCD-05,MYUCD-06　　2,000円
MYU-RECODINGS　発売中

音楽専門の草の根パソコン通信ネットワーク「MYU-NET」(03(5248)2821，24時間，9600bps，GuestID:0)が年に2回発行しているオリジナルコンビネーションアルバムが今年の夏も発売となった。

今回発売となった「MYU-TATION」は2枚組で全23曲収録。ボーカル曲もあるがテクノ系のインストルメンタルが中心の内容となっている。

また，これと同時に，同ネットのテクノユニット「ULTRA7」の1stソロアルバムである「TRANSLUCENT SOUND」もリリースされたとのこと。

これらの同人CDの購入希望者は上記ネットへアクセスするか，あるいはe-mail:mazda@ed.teu.ac.jpに問い合わせてみてほしい。

続・テレビの話

先月このコーナーで紹介したソニー製テレビモニタKX-29HV3のリモコン「隠しコマンド」だが，それと同じコマンドが同社のKV-21STII，KV-25STIIなどでも使用可能なことが判明した。

意外と，最近のソニーのテレビはみんな同じコマンドが利いちゃうのかな。

新Xの話

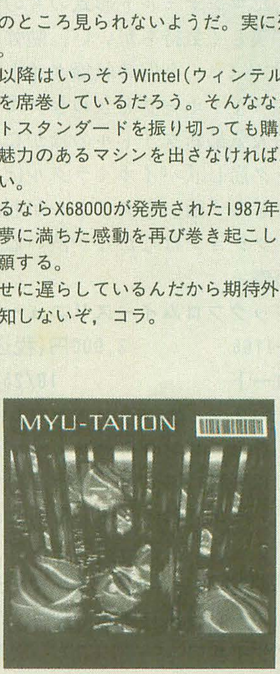
今年の11月には日本でもWindows95が発売される。マイクロソフトの消費者を煽り立てるような宣伝力で，Windows95は日本でもきっとよく売れることだろう。そして，この11月のマイクロソフトによる戦略的大イベントにあわせて，年末あたりに大々的なマシンの買い換えが行われることは想像にかたくない（「Windows95は以前より高速になった」という記事と「Windows95での標準CPUはPentiumだ」という2種類の記事が同時に雑誌上に見られるのはなぜだろう？）。

だから私としては，シャープのX68000ユーザーに対する正式なメッセージがこのイベント前にほしかったと思う。しかし，そういった動きはいまのところ見られないようだ。実に残念なことだ。

来年以降はいっそうWintel(ウィンテル)パワーが世を席巻しているだろう。そんななか，デファクトスタンダードを振り切っても購入したくなる魅力のあるマシンを出さなければ，成功は難しい。

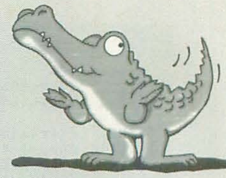
できるならX68000が発売された1987年，あのときの夢に満ちた感動を再び巻き起こしてほしいと懇願する。

遅らせに遅らしているんだから期待外れだったら承知しないぞ，コラ。





(善)のゲームミュージックでバビンチョ



西川善司

●ナムコゲームサウンドエクスプレス
Vol.21　グレートスラッガーズ
CD:VICL-15045　1,500円(税込み)
ビクターエンターテインメント　発売中

野球ゲームのBGMということで応援歌中心か思いきや，実にかっこいいカシオペア系のフュージョンが満載。特に3曲目の「Great Sluggers」はカシオペアの「intensive way (Freshness)」を彷彿させる曲の構成/展開で，ファンなら思わずニヤリ。

そのほかの曲もゲームミュージックフュージョンにしては珍しくギターがリードをとるタイプのものが多く，なにかとても新鮮で爽快な印象を受ける。惜しむらくはアンビエンス系のエフェクトが皆無であること。聞くときにはDSPなどで軽くエフェクトをかけて楽しみたいところだ。

・おすすめ度　10

●ウィンビーのネオシネマ倶楽部3
-ときめき編-
CD:KICA-7674　3,000円(税込み)
キングレコード　発売中

数あるコナミのゲームミュージックをアトランダムにチョイスし，イージーリスニング系にまとめ上げたこの「ネオシネマ」シリーズも今回で3弾目。全体的にピアノを中心に据えたコンチェルト形式のものが多く，静かで美しく気持ちがいい。勉強や読書のBGMにもいい。今回の収録タイトルは「ツインビーヤッホー」「ときめきメモリアル」などの人気最新タイトルのほかに「謎の壁ブロック崩し」「バイオミラクルぼくってウパ」などの「通」なタイトルが収録。今回のセレクションはなかなか渋いね。

・おすすめ度　9

●ミュージックフロムイースリニューアル
CD:KICA-1168　3,000円(税込み)
キングレコード　10/25発売

イース1のオリジナルサントラはいまだに売れ続けているロングセラー。今回，これを音質的音楽的にグレードアップしたアルバムがリリースされることになった。

生バンドでもない，オーケストラでもない，余計な展開もない，原曲のFM+PSGサウンドのイメージ継承に重点の置かれた純粋なシンセサイザミュージックだ。SEを除いたイース1の全34曲(未使用曲も含む)が収録されている。

ファルコムは同社のゲームソフト「ザナドゥリニューアル」のようなリニューアルプロジェクトをゲームミュージックにも展開していくのかもしれない。

・おすすめ度　8

●スーパーボンバーマン3
Original Sound Track
CD:PSCN-5025　2,000円(税込み)
NTT出版　発売中

独特の音階とリズムが印象的なボンバーマンの曲達。なんかアラビア的なイメージが香る不思議なサウンドだなぁと思っていたら，作曲者である竹間淳氏はアラブ音楽の専門家，自らもアラビアンフルート「ナイ」の著名な奏者であらせられた。

今回のスーボン3の曲たちも基本的にあのアラビックなテーマをあの手この手でアレンジしたものが中心。しかし'90年代が生んだ流行のテクノリズムとの見事な融合により，ゲームミュージックの枠を超えたともいえるまったく新しい音楽に仕上がっている。

・おすすめ度　9

●ナムコゲームサウンドエクスプレス
Vol.23　サイバーサイクルズ
CD:VICL-15047　1,500円(税込み)
ビクターエンターテインメント　10/21発売

ナムコのバイクゲーム「サイバーサイクルズ」のオリジナルサントラ。バイクのゲームというイメージからなのか，アウトローなイメージのハードロックが中心。

ちょっと和音の気持ち悪い曲もあったりするがそのへんはご愛敬。私の知る限り，ナムコ初挑戦の本格HRなのかな？　とにかく新ジャンルを展開したナムコの今後に期待。

・おすすめ度　6

●ナムコゲームサウンドエクスプレス
Vol.24　レイブレーサー
CD:VICL-15048　1,500円(税込み)
ビクターエンターテインメント　10/21発売

ナムコが放ったドライブゲームの革命児「リッジレーサー」の続編「レイブレーサー」のオリジナルサントラが早くも登場だ。もちろんBGMは「リッジ」を継承したあのノリのテクノハウス。メロディを楽しむよりは音の配置配列の巧みさやスピード感を味わう音楽の聞き方ができないとちょっときつめの内容だ。ただしドライビングBGMとしてはエンジン音との相性が絶対的によい。これを聞いて車の運転をするときはくれぐれもスピードの出しすぎには注意して。

・おすすめ度　8

●ときめきメモリアル
オリジナルゲームサントラPlayStation版
CD:KICA-7675～7676　3,800円(税込み)
キングレコード　10/25発売

いまだゲームのほうは発売されていないが(9月末現在)，サントラのほうの発売が決定した（ゲームは10/13発売予定)。2枚組構成のアルバムで収録曲はなんと全116曲。ゲームが，高校生の1年間を題材にしている関係で学校生活の1年のさまざまな行事のテーマ，数々のデートスポットのテーマがメロディアスなコナミ節で綴られている。365日を音楽で表現した生活事典といった感じだ。116曲がどんな画面で流れるのか，いやがうえにもゲームへの期待が高まる。

・おすすめ度　8



ミュージックフロムイースリニューアル



ときめきメモリアル

Assembler

ドライブ，ガコガコいってませんか?

Murata Toshiyuki 村田 敏幸

パソコンを使っているとファイルのコピーや削除は頻繁に行います。もし，これが高速化できるとしたらどうでしょうか? 今回は遅延書き込みを利用したツールとデバイスドライバを解説します。もちろんファイル圧縮ツールの展開にも効果があります。

先日，僕はRAMディスク上に作成した何10本かの小さなファイルをフロッピーディスクに落とそうと，愛用の(cpというありがちな名前の)自家製ファイルコピーツールを起動した……つもりだった。なのに，どういう無意識の悪戯か，指は“cp”ではなく“copy”とタイプしていて，もう久しく使っておらず，半ばデッドコードと化していたCOMMAND.X内部コマンドのCOPYを，そうとは知らずに呼び覚ます結果になった。墓場から蘇ったゾンビよろしく，COPYは僕を驚かせ，少々怖がらせもした。実行した途端響き渡る，がこがこがっこん，という耳障りな音。てっきりドライブの機械的な異常だと思った僕は「わわっ」とかなんとか間抜けな声を上げ，咄嗟にCTRL＋Cを押していた。

数瞬後，画面を見て自分の間違いを知り，事態を把握するにつれて，当初の狼狽は，ばつの悪さ交じりの安堵感に変わった。異常に思えた音が，実は，まるっきり正常な作動音に過ぎなかったことに気づいたからだ。要は相対性の問題，慣れの問題だった。僕のcpは，もうちょっとばかり静かに(そして，もうちょっとばかり速く)動いてくれるのだ。

＊

というわけで今回は，件のcpを初めとする僕の秘蔵(でもなんでもないが)自作ツール群で共通に使われ，その速度性能を高めるとともに静音効果まで与えている魔法(でもなんでもないが)のルーチンを，任意のプログラムから利用可能なライブラリとして提供する。こいつは一種の逆ディスクキャッシュを実現するサブルーチン集で，“若干のリスク”と引き換えにディスクへの書き込み性能を向上させる。より正確には，書き込み回数をできるだけ減らし，それに伴ってシーク回数を減らし，これらの相乗効果で処理時間を短縮する。シーク回数が減る分，ドライブの動作音は小さくなり，稼動部分の摩耗も少なくなって機器寿命も延びる(かもしれない)，というオマケももれなくついてくる。

高速化の対象となるのはHuman68kレベルでのセクタサイズが512あるいは1024バイトのディスク装置すべてで，書き込みとシークを減らすという動作原理上，フロッピーディスクのような遅い装置ほど，より顕著な効果が期待できる。逆に，RAMディスクには効かないどころか，オーバーヘッドが増える分だけ遅くなる。もっとも，メーカー純正(あるいは準拠)のRAMディスクは自動的に処理対象から外すようになっているので，悪影響を受けるのは変則的なメディアIDを持つRAMディスクに限られる。

応用は枚挙にいとまがない。たいていの書き込みを伴うファイル操作は，このルーチンを組み込むことによって，多少なりとも速く静かになると思ってもらっていい。とりわけ，ファイルの削除，リネーム，移動，タイムスタンプやファイル属性の変更，といったFATとディレクトリの書き換えのみからなる操作との相性は抜群で，ときに数～20倍程度もの劇的な速度向上が見られることもある。通常のファイル出力の場合はそこまで極端ではないものの，多数のファイルに対して書き込むほど，また，出力データ量が少ないほど効果が高まるという性質があるため，複数ファイルのコピー，巨大ファイルの分割(いわゆるsplit)，アーカイブの展開など，ある程度以上まとまった数の中小ファイルを作成する操作にならそれなりに効く。数10%増しの速度は出てふつう，2倍速前後になることも珍しくはなく，RAMディスクからフロッピーディスクへ小さなファイルを大量にコピーするような都合のいい場面であれば3倍速以上もあり，だ。

既存プログラムへの組み込みも容易で，基本的には，初期化とあと始末に対応する2つのサブルーチン呼び出しを追加するだけでよく，メインの処理に手を加える必要はない。アセンブリ言語レベルのみならず，Cプログラムからの利用も可能だ。また，効果は子プロセスにも伝播するので，メーカー純正コマンドのようにソースがないプログラムも，“外づけブースター”を用意すれば無改造のままで高速化できたりする。

組み込むことでの実行ファイルサイズの増加も180バイト程度とごく小さい。実は，ある理由によ

り，処理本体はデバイスドライバ形式でメモリ上に常駐する仕様になっており，ユーザープログラムには常駐部とのインタフェイスルーチンのみがリンクされるのだ。常駐部も1.5Kバイトほどだから，メモリを圧迫することはないだろう。

でもって，忘れちゃいけない“若干のリスク”だが，ひとことでいってしまえば，人災と天災に脆い。操作ミス，あるいは，停電などの事故が起きると，最悪の場合，ディスクの管理領域に致命的な不整合が生じて多くのデータが失われうる。いや，ダメージの大きさだけを見れば，世の中にはもっと危険なプログラムだってあるわけで，フラグメンテーションを起こしたディスクをきれいに整理するツールなんかはその筆頭だ。が，その種のプログラムの使用頻度は決して高くなく，しかも，使うときには十分な注意を払って事前にディスクをバックアップするぐらいの安全策はとるのに対して，今回のルーチンは日常的に頻繁に使われるプログラムでの利用を想定しているため，より危なっかしいともいえるのだ。ファイルをコピーしたり削除したりする前にいちいちディスク全体のバックアップをとる人は，まず，いないに違いない。

また，Human68kの未公開情報に依存していることもあって，相性の悪い常駐プログラムやデバイスドライバも存在するかもしれない。互いに足を引っ張り合って処理速度を落とすぐらいならまだいいが，致命的な結果を招くことも十分考えられる。

加えて，バグもないとはいい切れない。一応，オリジナルは3年半ほどの“テスト期間”を無事故無バグで切り抜け(最初のバージョンが動くようになるまではいろいろあったけど)，Human68k ver. 3に対応させるためのやや大規模な改修からでも2年は経っているが，環境の変化によって潜在的なバグが表面化する危険性はあるし，掲載にあたって若干手を入れたことでエンバグした可能性だってなくはない。ペーパーメディアで提供する以上，打ち込み間違いも生じるだろう。いずれにしろ，非常にクリティカルなプログラムだから，バグったが最後，何が起こるかはちょっと予想できない。書き込み対象以外のドライブも安全ではなくなるのだ。

繰り返しになるが，今回提供するのは“リスクと引き換えに時間を節約する”ルーチンであり，節約されるのがユーザーの時間である以上，リスクもまたユーザー持ちとなる。万が一のことがあってもこちらでは一切保証しかねるので，利用するかどうかは各自の責任において慎重に決めてほしい。

なお，実行にはHuman68k ver. 2以降が必要となっている。この点もまた了承願いたい。

DELが遅い

ことの起こりは，DELコマンドだった。MS-DOSのCOMMAND.COMのDELなら，ファイルをひとつ消す場合でも，ワイルドカードを指定して複数ファイルをまとめて消す場合でも，処理時間に実質的な差はない。ところが，COMMAND.XのDELは，削除するファイルの数に比例して処理時間が長くなり，ドライブの性能などによっては欠伸が出るほど待たされることがある。Human68kはMS-DOSと同じディスク管理方式を採用しているというのに，どうしたらこんな差が出てくるのだろう？

ファイル削除の手順を知っていれば，両者の内部動作がどうなっているかは，容易に想像がつく。念のため，MS-DOS/Human68kにおける1ファイル削除の手順を簡単にまとめておこう。

1） ディレクトリ領域を読みつつ，削除対象ファイルに対応するディレクトリエントリを探す

2） 見つけたディレクトリエントリから，ファイルが収められている先頭クラスタ番号を取り出し，対応するFAT領域を読み込む

3） FATを辿りながら，ファイルが収められているクラスタに対応するFATエントリを，“空きクラスタ”の印に順次置き換え，更新結果をディスクに書き戻す

4） ファイルに対応するディレクトリエントリの先頭を“削除済み”の印に置き換え，更新結果をディスクに書き戻す

大雑把にいえば，FATとディレクトリを読んで，更新して，書き戻す，だ。

明らかに，COMMAND.XのDELは以上の手順をファイル1本ごとに繰り返している。FAT領域もディレクトリ領域も複数のファイルで共用されている(注1)から，いくつものファイルを続けて削除する過程では，たったいま書き込んだばかりのFATとディレクトリを読み込み直して書き戻す，といった重複したディスクアクセスが行われるわけだ。このうち読み込みの重複については，CONFIG.SYSのBUFFERSで確保されるHuman68k内部のバッファがキャッシュとして働くので問題にはならないが，書き込みだけを見てもかなりの無駄といえる。しかも，FATとディレクトリを交互に更新する点に注目してほしい。このため，ディスク上での両者の位置が離れていると，ドライブのヘッドが行ったり来たりを繰り返すことによって，さらに夥しい時間が浪費される。シーク(ヘッドの移動)はディスクの読み書きそのものよりもずっと時間がかかるのだ。

と，ここまで書けば，COMMAND.COMのDELの動作もまた明らかだろう。要するに，いったんFATとディレクトリを読み込んだらメモリ上で複数ファイルの処理を行い，最終結果だけをディスクに書き込んでいるわけだ。

ちなみに，この違いはCOMMAND.XとCOMMAND.COMの作りの違いというよりも，Human68kとMS-DOSのシステムコールの差に根差すものだ。Human68kのDOSコールdeleteは指定されたファイルただひとつを削除する仕様であり，それで完結した機能単位である以上，1ファイル削除するごとにFATとディレクトリを更新せざるをえない。対する

注1）1セクタが1024バイトの場合，ディレクトリ領域1セクタには32ファイル分のファイル情報が格納される。また，FAT 1セクタには最大で800個近くのクラスタについての情報が格納されるので，クラスタサイズ未満のファイルばかりなら，同数のファイルによって共用されることになる。

MS-DOSにはファイル削除のファンクションコールが2つあり，一方はdelete相当だが，他方はワイルドカードに対応し，複数のファイルをまとめて削除する機能を持っている(注2)。こちらは“複数ファイルの削除”という機能単位だから，FATとディレクトリの更新を一括して行うことが許されるのだった。

さて，Human68k上で複数ファイルを削除する際には，本来必要な回数よりも，ずっと多くのディスクアクセスが行われていることがわかってもらえたと思う。この無駄を省けば，より高速なファイル削除ツールが実現できるはずだ。ただ，やっかいなことに，重複したディスクアクセスはファイル削除処理の要であるDOSコールdeleteの中，というか，deleteを続けて呼び出す一連の手順の中に埋もれている。となれば，deleteを使わない方向で考えるのが成り行きというものだろう。つまり，DOSコールdiskredとdiskwrtによるセクタ単位の読み書き機能を使って，FATとディレクトリを直に書き換えるというアプローチだ。

僕の頭に最初に浮かんだのも正にこの方法だった。ただし，あくまで第2候補，ほかに手段がないときのための予備としての採用だ。実はHuman68k ver.3ではすでに改善されているのだが，僕がこの問題に取り組んだ当時のdiskredとdiskwrtには，MS-DOSから意味もなく受け継がれた(意味があったのならあとで修正するはずがない)仕様上の欠陥があり，できることなら使わずに済ませたかったのだ。それというのも，ver.2までのdiskredとdiskwrtはHuman68k内部のバッファの存在を無視するため，これらのコールを利用するとバッファのキャッシュ効果が損われ，本来はしなくても済んだディスクからの読み込みが発生するからだ。とくにdiskwrtを使うとバッファとディスク内容が一致しなくなる(注3)ことから，DOSコールfflushで全バッファを破棄しなければならなくなり，そうなると破棄したバッファの内容はあとで必要になったときに再び読み込まなければならず……と“風が吹けば桶屋が儲かる”式に大量の読み込みが発生して，計算上，その数は節約できる書き込み回数を楽に上回る(注4)。

では，どうするかだが，この問題の解決法としては，無駄を避けるか，無駄を潰すか，のどちらかしかありえない以上，次に検討すべきは，無駄を潰すこと，つまり，deleteを利用しつつもdeleteが無駄を生じないようにすること(!)だ。当初，そんなことが可能かどうかは半信半疑だったのだが，しばらく発酵期間を置くうちに，ひとつのアイデアが浮かんだ。もっと早くに思いつかなかったのが不思議なほど，単純で，それでいて，ファイル削除以外にも応用可能な，強力な方法だ。

遅延書き込みの採用

基本となるアイデアは，マルチタスクOSにしばしば見られる“遅延書き込み(write-behind：あと書き)”だ。これは，文字どおり，ディスクへの書き込み要求をその場では処理せずに，いったんメモリ上にプールしておいて，あとで一括して書き出す手法をいう(注5)。

一般論として，書き込みを遅らせると何が嬉しいかといえば，第1に，ディスクドライブのヘッドの移動距離を小さくできる。通常，ファイル本体を収めるデータ領域とその管理情報を収める領域はディスク上では離れていて，管理情報自体も分散されている(MS-DOS/Human68kならFATとルートディレクトリと複数のサブディレクトリ)ものだから，一連のファイル処理の過程では，ヘッドは結構あちらこちらへと移動して回る。とくにマルチタスク環境となると，各タスクそれぞれがディスクの別々の領域にアクセスするため，これをリアルタイムに処理しようとすれば，ヘッドは引っ切りなしに跳ね回ることになる。しかし，ある程度の回数の書き込み要求を溜め込めば，ヘッドの移動が最小になるよう，書き込み順序を最適化する余地が生まれるわけだ。

第2に，ディスクの回転待ち時間を減らすことができる。ディスクは連続する領域に続けざまにアクセスするときに最高速度が出るよう最適化されている(少なくとも，そうされているべき)ものであり，同じ連続セクタにアクセスする場合でも，各セクタのアクセスの合間にほかの処理を挟んで変に時間を空けたり，あるいは，ひねくれた順序でアクセスしたりすると，余分な時間がかかる。というのも，アクセスしようとするセクタがひとたびヘッドの下を通過してしまえば，ディスクがもう1回転するのを待たなければならないからだ。この点，書き込みを遅延すれば，飛び飛びの時間間隔，飛び飛びの順序で発生した連続領域への書き込み要求をまとめることが可能になる。

第3に，ディスクの同一領域に対する重複する書き込みとそれに伴うシークを排除することができる。遅延書き込み時には，同じ領域に対して何度書き込み要求が発生したとしても，中間結果はメモリ上に蓄えられるだけなので，実際にディスクに書き込むのは最後のただ1回で済むのだ。この性質は，いまの目的にうってつけといえる。

逆にデメリットとしては，安全性の低下が挙げられる。書き込みを遅らせる以上，最終的な書き込みが行われる前に，ユーザーが誤ってディスクをイジェクトしたり，電源を落としたり，リセットしたり，あるいは，停電が起きたりすると，当然，書き込まれなかったデータは失われる。タイミング次第ではディスクの管理情報に不整合が生じ，より多くのデータが失われることもあるだろう。だが，得られる速度性能を考えれば，多少の危険は冒してみる価値があるように思えた。そこで，さっそく実装に向けて検討に入ったわけだ。

まず，どのレベルで遅延書き込みを実現するかだが，いま問題にしている重複したディスクアクセス

注2）その代わり，カレントディレクトリ上のファイルしか削除できないとか，決められた構造を持ったメモリ領域にファイル名をセットしてから呼び出さなければならないといった使いにくさがある。

注3）Human68k ver.3ではこの点が改善されたため，diskwrtを使うからといってfflushを発行することは必須ではなくなっている。このためなのだろう，fflushを無効化する方法が用意されていたりもする。

注4）それなら，第2候補にも値しないようなものだが，バッファへの再読み込みの大部分はdiskwrt(とfflush)を使ったプログラムの終了後に少しずつ行われるため，そのプログラム自体の処理速度には影響しない。あとでツケを払うことにはなるのだが，いわば分割払いになることもあって，気にしなければ，そう大きな問題にはならないのだ(僕は気にするが)。

注5）これと対になるのが“先行読み込み(read-ahead：先読み)”で，こちらは読み込み要求が発生したときに実際の読み込み指定量よりも多めにディスクを読んでおいて，次回以降の要求時にはここから切り出すという手法だ。

はDOSコールの中で発生することから，事実上，選択肢はひとつしかない。遅延書き込み処理部は，Human68k本体とデバイスドライバとのあいだに割り込ませる。より正確には，すべてのブロック型デバイスドライバの頭の部分に取りつき，Human68kからのI/Oリクエストを横取りする。そして，デバイスドライバに渡されるはずだったI/Oリクエストコマンドを調べ，セクタ出力コマンドであれば，その出力データを内部のバッファに溜めて，何食わぬ顔で呼び出し元のHuman68kに戻る。セクタ入力コマンドで，該当セクタがバッファにあればその内容を転送してから，やはりHuman68kに直接戻る。該当セクタがないか，それ以外のコマンドの場合は，デバイスドライバの元のエントリに制御を渡し，ふつうに処理させる，という段取りだ。

ただ，Human68kはディスクに関するあらゆる入出力をデバイスドライバ経由で行うわけだから，遅延書き込みモジュールがデバイスドライバに取りつくと，いま目的としているファイル削除ツールだけではなく，メモリ上のあらゆるプログラムが遅延書き込み環境で動作することになる。それを見越すと，遅延書き込みモジュールは汎用かつ堅牢な作りにしておかなければならない。仮に，削除ツールの実行中にのみ遅延書き込みが有効となるように限定したとしても，当のツールがリダイレクトされているかもしれないし，バックグラウンドプロセスがディスクを読み書きするかもしれないから，一般のファイル出力を含む，削除ツールならやりそうもない操作にも対応する必要がある。もし，ファイル削除の処理以外ではディスクの読み書きが発生しないという保証があれば，読み書きされるのはFATかディレクトリで，対象となるメモリはHuman68kのバッファだという仮定が成り立つので，入出力要求量はつねに1セクタ，入出力先バッファの先頭は偶数アドレス，といった決め打ちができるのだが，そのような簡略化は許されないということだ。

逆に，汎用に作成しておけば，削除ツール以外にも応用が利くようになる。リネーム，移動，タイムスタンプやファイル属性の変更などの操作も，複数ファイルに適用すれば，やはり重複してFATとディレクトリが更新されることになるし，比率としては少ないものの，複数ファイルのコピー時など，いくつものファイルを生成する際にも同様の冗長さは見られる。ちゃんとした遅延書き込みモジュールを用意しておけば，ファイル削除どころか，たいていの書き込み操作を高速化する道が開けるのだ。

で，こうなると，遅延書き込みモジュールをデバイスドライバなり常駐型プログラムなりにして，つねに遅延書き込みが有効となるようにすることも考えられる。遅延書き込みバッファのフラッシュ(flush：掃き出し)は，実際にマルチタスクOSがそうしているように，バックグラウンドプロセスによって定期的に行えばよい。だが，さすがにそれはやめておいた。プログラム開発時には，しばしばリセットしなければ回復できないような致命的なバグも出るわけで，バッファがフラッシュされていないのを知りつつ，涙を飲んでリセットスイッチを押さなければならない状況に陥ることも考えられる。それはぜひとも避けたい。また，個々のプログラムの動作は，とりあえずそれ自体で完結していてほしいようにも思えた。たとえば，「n個のファイルをコピーしました」といった類の終了メッセージが表示された時点では，実際にすべての書き込みが完了していてくれたほうが安全だろう。そこで，遅延書き込みは基本的に個々のプログラムの動作中のみ有効となるようにし，バッファのフラッシュも明示的に行う，という線を守ることに決めた。

個々のプログラムが遅延書き込みモジュールの登録/解除を行うとなると，多重登録に対する配慮も必要だ。すでに遅延書き込みが有効な状態で，さらに遅延書き込みモジュールを割り込ませたのでは，処理が重複して速度が低下するばかりか，バックグラウンドプロセスの存在やSX-WINDOW環境を想定すると危険ですらある。登録と逆順に解除される分にはよいが，この順序が乱れると，失われるデータも出てくるだろう。これを避けるためには，セマフォを設けて，多重登録を禁止するようにしなければならない。で，詳しくはコラムに譲るが，最終的に，遅延書き込みモジュールはメモリ上に常駐しているべきだという結論に達した。上で常駐型にはしないと決めたのに反するようだが，ここでいう常駐は“メモリ上に置いておく”というだけの意味で，通常はHuman68kのファイルシステムからは切り離された状態を保つ。そして，外から内部サブルーチンを呼び出す方法を用意しておき，ユーザープログラムがこのサブルーチンの呼び出しによって遅延書き込み開始を指示した時点で，遅延書き込み処理本体をデバイスドライバに取りつかせる。この際，内部的にセマフォをロックして“遅延書き込み中”だということを示すようにし，二重登録を防止するというわけだ。常駐するのは，登録/解除を一元管理するための手段に過ぎない。

と，このくらいまで煮詰めておけば，ぼちぼちとプログラムの制作に取りかかれそうだ。細部については，完成した実物のプログラムを見てもらいながら説明することにしよう。

フグは食いたし命は惜しし

リスト1のSWELLFISH.Sが遅延書き込みを実現するデバイスドライバ型の常駐ライブラリだ。利用するには，

```
AS SWELLFISH
LK SWELLFISH /OSWELLFISH.SYS
```

のように単独でアセンブル/リンクしたうえで，得られたSWELLFISH.SYSをCONFIG.SYSのDEVICEで登録すればよい。

では，さっそくリスト1の中身を順に追っていく

ことにしよう。

16～42行ではこのプログラムが使ういくつかの構造体の定義をしている。16～20行のデバイスドライバヘッダについてはいいだろう。続く25～42行は，Human68k内部で各ドライブについての情報を保持している内部DPBの構造定義だ。本来，内部DPBはユーザープログラムで利用すべきものではないのだが，このプログラムでは必要に迫られて，システムの領分に土足で踏み入っている(このプログラム自体がそうだという話もある)。DOSコールgetdpbで得られるDPBとは構造が異なることに注意してほしい。以前にも触れたように，Human68k ver. 1ではgetdpbで得られるDPBとHuman68k内部のDPBは同じ構造をしていたのだが，ver. 2からはこのような構造に変更されている(注6)。

44行はRAMディスクのメディアID定義。RAMディスクは，遅延書き込みによって書き込み回数を節約したとしても，それに伴って省略できるシークが最初からないため，速くならないどころか，オーバーヘッドが増える分だけ遅くなってしまう。そこでリスト1では，RAMディスクは処理対象から除外する(RAMディスクドライバには取りつかない)ようにしてあり，その識別に使うのがここで定義しているメディアIDだ。

49～57行は各デバイスドライバのストラテジエントリに引っかけるコードの構造定義だ。先ほどから

注6）Human68k ver. 3ではさらに内部DPBが拡張され，ver. 2の構造の後ろにもう26バイトがくっついているのだが，FAST～.X用らしいということしか調べがついていない。

セマフォ

セマフォ(semaphore)とは，複数プログラムが同一のコンピュータ資源を取り合って競合するのを防ぐ，いわゆる排他制御のためのメカニズムで，元々は鉄道の腕木式信号を指す言葉だ。鉄道のセマフォが，1本の線路に2本の列車が進入して衝突するのを防ぐ目的で片方の列車をブロックするように，コンピュータ内のセマフォはプログラムが使用中の資源をほかのプログラムからは使用できないようにブロックする。といってもセマフォの実体は単なる変数であり，通常は，ある資源が使用中か未使用かを表すフラグに過ぎない(同一の資源が複数ある場合は，未使用資源数を保持するカウンタとなる)。各プログラムは特定の資源を利用したければ，セマフォを調べて，未使用だったら使用中の印にして(ロックして)から使い，使い終わったら未使用の印に戻す(アンロックする)。ただ，マルチタスク環境(この文脈ではHuman68kもその範疇に入る)では，セマフォのアクセス自体も排他的に行う必要があり，とくにセマフォがロックされていないかどうかの確認と，セマフォのロックとのあいだに，ほかのプログラムがセマフォへアクセスすることを禁止できなければならない。このため一般にはOSがセマフォへの排他的なアクセス手段を提供するのだが，その手段を持たないHuman68kでは，ユーザープログラムレベルで頑張ることになる。そこでHuman68k上でセマフォを実現する方法をいくつか考えてみた。

1) 任意の1バイトメモリを使う

68000にはセマフォ用ともいうべきtas命令があるので，どのプログラムからもアクセスできるメモリが1バイト確保できればセマフォが実現できる。tasは指定のメモリ1バイトを0と比較してccrに反映すると同時に，そのバイトの第7ビットを立てる命令だ。0をアンロック状態，非0をロック状態と決めれば，この1命令でセマフォの検査とロックが行える。ただ，肝心の1バイトをどこから捻出してくるかが問題だ。IOCSのワーク領域なり，SRAMなりの未使用バイトを使用する手はあるが，それでは同じ方法でセマフォを実現しているプログラムと競合する危険がある。安全のためには，DOSコールmallocで小さな識別用メモリブロックを確保したうえで，適当な識別用文字列をつけておき，このメモリブロック上の1バイトを使うべきだろう。Human68kのメモリ管理ポインタを辿りながら，識別文字列を頼りにセマフォ用に確保したメモリブロックを探し，見つけたらtasでセマフォをロックするわけだ。ただし，最初にメモリブロックを確保するタイミングに気をつける必要がある。ふつうに考えると，上の手順中，セマフォ用のメモリブロックを探す処理の結果，見つからなかったら確保する，というのが自然だが，この確保の瞬間にスレッドが切り替わるかもしれない。Human68kにおいてスレッド切り替えが行われないことが保証されているのは，スーパーバイザモード時で，かつ，DOSコールを発行しない期間だから，DOSコールmallocでメモリを確保すると，その直後，mallocから戻る前にスレッドが切り替わる可能性がある。メモリブロックは確保されているが，セマフォとしては初期化されていない状態でほかのプログラムに制御を渡したのでは，ちゃんとしたセマフォが実現できない。というわけで，やるとすれば，事前に一度，セマフォ用のメモリブロックを確保して初期化する専用プログラムを走らせることになるだろう。もし，このプログラムがマルチタスク的に実行されれば同じ問題が起こるが，そこまで考慮することはあるまい。

ちなみに，通常，mallocで確保したメモリブロックはプログラムの終了と同時に解放されてしまうが，メモリ管理情報中の“このメモリブロックを確保したプロセスのプロセスID”の欄を書き換えて，プロセスIDとしてはありえない値(たとえば，プロセスIDは絶対に奇数にはならない)にするなり，Human68k本体のプロセスIDにするなりすれば，解放されなくなる。

2) 特定のファイルの有無を使う

Human68kで最も排他制御がしっかりしている部分といえばファイルシステムだ。そこで，特定のファイルが存在するかどうかでロック/アンロックを区別するという方法が考えられる。DOSコールnewfileを使えば，ファイルが存在するかどうかの検査と新規作成をひとまとめで行うことが可能だ。だが，この方法は確実ではあるのだが，とにかく遅いという欠点がある。

3) 特定のキャラクタデバイスの有無を使う

2)に似ているが，代わりに，キャラクタデバイスを使う。つまり，特定の名前のキャラクタデバイスをオープンしてみて，オープンできなかったら，着脱可能なデバイスドライバと同様の手法により，そのキャラクタデバイスを登録する。ただ，こうして実現したセマフォは状態の検査とロックとのあいだにタイムラグが生じるため，このあいだにスレッドが切り替わったりしないよう，手を打たなければならない。

4) コモン領域を使う

Human68k ver. 2からサポートされながら，ひょっとすると誰も使っていないのではないかと思わせるマイナーな機能にコモン領域というのがある。これはHuman68k内部に確保された共有メモリで，各プログラムはDOSコールcommonを通じて，この共有メモリ上に名前つきのメモリブロックを確保したり，そのメモリブロックの内容を読み書きすることができる。イメージとしては，メモリ上に小さなファイルを作るようなものだ。このコモン領域により，少々原始的ではあるが，プロセス間通信が実現できる。コモン領域上のメモリブロックは特定のプロセス以外からはアクセスできないようにすることも可能だから，セマフォにも利用できそうだ。と思ったのだが，実際には，コモン領域上のメモリに対して直にアクセスできるわけではないのでtas命令は利用できないし，読み込み操作と書き込み操作が分離されているため，アンロック状態かどうかの確認とロックとをまとめて行うことができない。しかも，アクセスにはDOSコールを利用するわけだから，このときスレッドが切り替わる可能性があり，セマフォへの排他的なアクセスが保証されないことになる。どうもこの方法はうまくないようだ。

5) セマフォに対する専用のアクセス手段を設ける

要するに，セマフォを腹に抱えたプログラムを常駐させておき，適当な方法でこのプログラムの内部ルーチンを呼び出して，セマフォのロック/アンロックを行う方法だ。なんなら，識別番号なり識別名なりで区別される複数のセマフォを管理する機能を持った専用プログラムを用意し，排他制御が必要なあらゆる用途で共用することもできる。

もっとも，プログラムの多重起動の防止目的で使うだけならば，多重起動されては困るそのルーチン自体が常駐して，セマフォはあくまで内部的なフラグとし，そのルーチンが呼び出されるときにこっそり参照するというスタイルのほうがよいだろう。当然のことだが，多重起動できないルーチンを各プログラムがばらばらに内蔵するから管理が面倒になるのであって，一元管理してしまえば話はずっと簡単になる。結局，今回の遅延書き込みモジュールでは，この方法を使うことにした。

注7)デバイスドライバコマンドの種類や引数を収めたメモリ領域を指すHuman68k用語。本当はコマンドパケットと呼んだほうがぴったりくるのだが。ちなみに，MS-DOSにおけるリクエストヘッダとは，言葉どおり，I/Oリクエスト時のコマンドパケットのヘッダ部分を指す。ところで，『Human68kユーザーズマニュアル』における“パス名”の定義は，UNIXやMS-DOSのそれとは違ってること，みんな気づいてる？

注8)プログラム自身がプログラムを生成して実行できるというのは，ある意味でマシン語の一番おいしいところかもしれない。もっとも，これは一種の自己書き換えであり，命令キャッシュを備えていながら，キャッシュとメインメモリとの不整合を検出できないプロセッサ(たとえば，MC68030)では注意しないと，うまく動かないことがある。ただし，リスト1はX68030でも問題なく動く(よほど変な使い方をされれば別かもしれないが)。

簡単にデバイスドライバに取りつくといってきたが，そのためには，デバイスドライバの2つのエントリ，ストラテジエントリと割り込みエントリの両方を乗っ取ることが必要になる。歴史的な理由(つまり，MS-DOSがそうだから)により，Human68kのデバイスドライバ呼び出し手順は，まず，リクエストヘッダ(注7)の先頭アドレスをa5レジスタに入れてストラテジエントリを呼び出し，デバイスドライバ側はこのa5を内部のワークに覚えておいて一旦Human68kに戻り，それを受けてHuman68kがデバイスドライバの処理本体である割り込みエントリを呼び出す，という2段構成になっている。遅延書き込み処理部本体は割り込みエントリに取りつくことになるが，それだけではリクエストヘッダが取得できないため，ストラテジエントリにも手を伸ばさざるをえないわけだ。で，このストラテジエントリに引っかけるコードはデバイスドライバの数だけ(理論上は最大26個)要り，しかも，リスト1では100バイトほどのテーブルとセットにする必要があったため，これをプログラム中に埋め込んだのではプログラムサイズが肥大化してしまう。それを嫌ってリスト1では，49〜57行のような構造のコードを内部的に必要な数だけ複製するようになっている(注8)。49〜57行自体は構造を定義しているだけで，複製時に使う雛型は，テーブル部分を除いた形で98〜201行に用意している。

ここで，49〜57行中，ところどころ0になっている箇所には，複製の際，適当なアドレスがはめ込まれる。具体的には，50行と55行の，

```
jmp    0.l
```

は，それぞれ，取りつくデバイスドライバの元の割り込みエントリとストラテジエントリへのジャンプに，また，54行の，

```
move.l #0,dpbtable
```

中の0は，51行のラベルORGINTENTに対応する“実物のコードにおけるアドレス”に置き換わる。49行の.offsetのオフセット初期値が0ではなく−2になっているのは，あとでこのORGINTENTを基点として相対的な参照をするための補正だ。

なお，割り込みエントリ側に取りつく遅延書き込み処理本体は，さすがにデバイスドライバの数だけ用意するわけにはいかないので，全デバイスドライバで共用するようになっている。この際，ストラテジエントリ側でワークdpbtableに収める，ORGINTENTに対応するアドレスが，どのデバイスドライバから飛んできたのかの識別用に使われる。

プログラム本体に入って，64〜70行。リスト1はデバイスドライバとして組み込まれるので，プログラム先頭はデバイスドライバヘッダから始まる。リスト1の場合，メモリに常駐する方便としてデバイスドライバの形式を借りるわけで，この場合，デバイス名にはファイル名としては無効な名前を使うのが定石だ。しかし，70行を見てのとおり，このプログラムは有効なデバイス名@FUGUを持っている。当然，この@FUGUはCONなどと同様に読み書きすることができる。ただし，デバイス属性としてNULデバイス属性を与えてある(65行)ので，読んでも何も返さず，書いても出力データは単に捨てられる。

こんな仕様にしたのは，DOSコールioctrlを利用して外部とのインタフェイスをとりたかったからだ。ioctrlはデバイスドライバの制御用に用意されたDOSコールで，デバイスドライバとユーザープログラムとのあいだで任意の情報をやり取りする手軽な手段となる。リスト1ではこれを利用して，内部ルーチンのアドレスをユーザープログラムが取得できるようにしている。で，ioctrlを使うためにはDOSコールopenでオープンできる正当なデバイス名が必要であり，といって，実際に入出力を行うわけではないのでNULデバイス扱いで十分ということなのだ。NULデバイスにしなくても，入出力コマンドの処理を空にすれば同じ結果が得られるが，Human68kはNULデバイスに対する読み書きを自動的に省略するので，NULデバイス属性さえ与えておけば，ダミーの入出力ルーチンを用意する必要すらなくなるという利点がある。実際にどのようにioctrlを利用するかについては，ずっと飛んで662〜707行にある@FUGUとしてのデバイスドライバ部を見てもらおう。

デバイス@FUGUはデバイスドライバに対するコマンドのうち，初期化(コマンド番号0)とioctrlによる入力(同3)のみをサポートしている。ただし，上述のようにNULデバイスとしたことで，自動的に入力と出力，および，入出力可能かどうかの検査がHuman68kによってサポートされる。

初期化部(695〜700行)は見てのとおり，タイトルを表示し，常駐する最終アドレスをHuman68kに伝えているだけだ。

アイデア上のポイントとなるioctrlによる入力部(676〜684行)も，処理自体は，4バイトの入力が要求されたら，内部ルーチンのジャンプテーブルの基準アドレスを返す，というごく単純な作りだ。

ジャンプテーブル自体は，90〜93行に用意してある。braを4つ並べた構成だが，braにワードサイズを指定していることに注意してほしい。これにより，ひとつのジャンプテーブルエントリが4バイトになることが保証される。したがって，ユーザープログラムは，ioctrlによって取得した基準アドレスに4の倍数を増減したアドレスを呼び出すことによって，リスト1がサポートする4つの内部ルーチン，

```
lazyupdate_calc
lazyupdate_open
lazyupdate_flush
lazyupdate_close
```

を利用することができる。基準アドレスがジャンプテーブル先頭から4バイトずれている(91行)ことから，仮に基準アドレスがa0に入っているとすると，上の4つのサブルーチンを呼び出すには，それぞれ，

```
jsr   -4(a0)
```

```
	jsr	(a0)
	jsr	4(a0)
	jsr	8(a0)
```

とする。ただし，リスト1はデバイスドライバとして常駐するので，ユーザーモードからはアクセスできないスーパーバイザ空間に置かれる。このため呼び出し時には，事前にスーパーバイザモードに移行しておくか，あるいは，Human68k ver.2からサポートされているDOSコールsuper_jsrを利用する必要がある(注9)。

なお，ジャンプテーブルの基準アドレスが先頭から4バイトずれているのは，ジャンプテーブル上のオフセットアドレスをaddqやsubqで扱える範囲に収めようとしたためだ。先頭＝基準ではオフセットの範囲が0～＋12になるが，4バイトずらせば－4～＋8だからaddqとsubqで届く。super_jsrを利用する際にはこっちのほうが都合がよいこともあるようだ。

さて，リスト1の内部ルーチン呼び出し手順は，スマートにまとまっているとは思うのだが，少々繁雑なのは否めない。そこで，より簡便に内部ルーチンが利用できるよう，別途インタフェイスルーチンを用意しておいた。これを使うと，リスト1の内部ルーチンがあたかもふつうのサブルーチンであるかのように呼び出せるようになる。まだリスト1の解説の途中だが，先にそちらを見てもらうことにしよう。リスト2のLAZYUPDATE.Sだ。

リスト2にはリスト1の内部ルーチンと同名のサブルーチン4つが含まれている。これらはリスト1の内部ルーチンへの橋渡しをするだけの小さなサブルーチンで，ジャンプテーブルの取得を含む一連の呼び出し手順をユーザープログラムになり代わって行う。各サブルーチンは頭に“_”をつけた名前でも外部定義されているので，Cからも同名の関数が利用できる。

リスト2自体はふつうの意味でのライブラリモジュールだから，

```
	AS	LAZYUPDATE
```

のように単にアセンブルしておき，生成されたリロケータブルオブジェクトファイルLAZYUPDATE.Oを任意のプログラムにリンクして使う。LAZYUPDATE.Oは，ほかのライブラリと同じディレクトリに置いておくなり，DOSLIB.Lあたりにまとめるなりすればよいだろう。

ここでリスト2に含まれるサブルーチン/関数(間接的にはリスト1の内部ルーチン)の機能と使い方を説明しておこう。

lazyupdate_calc(88～100行)は，遅延書き込みバッファとして何セクタ分確保したいかをロングワードの値で(ただし下位ワードのみ有効)スタックに積んで渡すと，リスト1から生成されるSWELLFISH.SYSが組み込まれていない場合は0を，そうでなければバッファ領域に必要なバイト数をd0.lに返す(Cの場合はこのd0.lの値が関数からの戻り値となる。以下同様)。バッファ領域自体は，この戻り値を使ってユーザープログラムが確保する。アセンブリ言語レベルならDOSコールmallocかsetblock，CレベルならmallocO関数あたりを使えばよい。ちなみに，バッファ数をNとすると，バッファ領域にはN×1036＋3440バイトが必要で，この数字の中には，前述のストラテジエントリに引っかけるコード(縁起ものだったので26個分)用の領域も含まれている。

lazyupdate_open(50～66行)は引数として，遅延書き込みバッファに使うメモリ領域の先頭アドレスとそのバイト数を渡すと，バッファを初期化し，遅延書き込みを開始する。通常，プログラムの起動直後，そのプログラムが行うディスク書き込みの前に呼び出せばよい。バッファ領域は必ずワード境界に整合している(＝偶数アドレスから始まっている)必要があり，X68030の場合はロングワード境界に整合させれば32ビットのバス幅が活かせて効率が上がる。戻り値としてはd0.lに，正常終了時0，SWELLFISH.SYSが組み込まれていないか，バッファ領域が小さ過ぎるか，すでに遅延書き込み状態のとき－1が返る。なお，リスト1側のlazyupdate_openの引数は3つで，上の2つに加えて，プロセスID(PSPアドレス)を必要とするのだが，リスト2は内部でDOSコールgetpdbにより自動的にプロセスIDを得るから，ユーザープログラムは気にしなくてよい。ここで，こうしてリスト2からリスト1へと渡されたプロセスIDは，遅延書き込み終了時の“鍵”とされるべく保存される。つまり，遅延書き込みを開始したプロセス(＝遅延書き込みバッファの持ち主)のみが遅延書き込みを終了することができるというわけだ。

lazyupdate_flush(68～77行)は遅延書き込みバッファの内容をフラッシュして，一旦バッファを空にしてから遅延書き込みを継続する。引数はなく，戻り値としてはlazyupdate_open同様の終了ステータスが返る。シェルやファイラーの類の対話型プログラムは遅延書き込み環境で動かすのが怖い(バッファのフラッシュを忘れてディスクを交換したり，電源を落としたりしてしまうかもしれない)が，処理の区切りごとにこのサブルーチンを呼び出すようにすれば，安全性が保てるだろう。

lazyupdate_close(79～86行)はバッファをフラッシュしたうえで，遅延書き込み状態を終了する。引数はなく，戻り値は0または－1だ。リスト1のlazyupdate_closeには先ほどの“鍵”としてプロセスIDを渡す必要があるが，例によってこの点についてはリスト2中で対処している。

ここで，lazyupdate_openを呼び出したプログラムは，終了前に必ずlazyupdate_closeを呼び出して遅延書き込みを終結させなければならないことに注意してほしい。そうしないと，プログラム終了後も遅延書き込み処理部が生き続け，終了したプログラムが確保した(＝すでに解放され，Human68kに返却された)バッファをそのまま使って遅延書き込みを継続してしまう。これではバッファ内容がいつ破

注9)super_jsrは任意のサブルーチン先頭アドレスをスタックに積んで渡すと，一次的にスーパーバイザモードに移行してから，そのサブルーチンを呼び出す。サブルーチンにはsr(含むccr)とsp以外の全レジスタがそのまま渡され，呼び出し元にはサブルーチン終了時のレジスタが返される(ccrは破壊される)。つまり，super_jsrを利用すれば，スーパーバイザ空間にあるサブルーチンをユーザーモードからでも呼び出すことができるわけだ。

壊されるかもしれず，非常に危険だ。アセンブリ言語レベルではDOSコールintvcsを，Cレベルではsignal()関数を使って，CTRL＋Cやハードウェアエラーによる中断時処理を適切に設定し，確実にlazyupdate_closeを呼び出してから終了するように細工しておく必要がある(アセンブリ言語での具体例はのちほど示す)。また，同じ理由で，遅延書き込みを利用するプログラムをデバッガ上で走らせる場合には，プログラムが終了する前にデバッガを抜けてはいけない。そもそも，まだ十分デバッグが済んでいないプログラムから遅延書き込みを利用すること自体危険なことだ。リスト2は，プログラムが完動するようになってから組み込むようにしてほしい。

リスト2には実際にはもうひとつユーティリティ的なサブルーチンが含まれている。このlazyupdate_exist(102～132行)は，例のジャンプテーブル基準アドレスを取得してd0.lに返す。SWELLFISH.SYSが組み込まれていなければ戻り値は－1になる。本来は，リスト2の内部ルーチンなのだが，SWELLFISH.SYSが組み込まれているかどうかの検査に応用できることから，外部からも利用できるようにしてみた(もっとも，lazyupdate_calcも同じ目的に使える)。

lazyupdate_existの中身はすでに述べたとおりのジャンプテーブル取得手順だが，多少エラーチェックが強化されている。まず，デバイス@FUGUを読み込みモードでオープンしてみる(107～110行)。オープンできなければSWELLFISH.SYSはまだ組み込まれていない。オープンできたら，DOSコールioctrlで装置情報を取得し，いまオープンしたのがSWELLFISH.SYSなのか，それとも，偶然存在する同名のファイルだったのかをチェックする(114～118行)。本物のSWELLFISH.SYSなら，ioctrl可能なNULキャラクタデバイス，という装置情報が返されるはずだ。ここまで一致すれば，本物とみなして大丈夫だろう。あとは，ioctrlによってジャンプテーブルの基準アドレスを得るだけだ(120～125行)。

遅延書き込み処理の実際

リスト1に戻って，各内部ルーチンと遅延書き込み処理本体の中身を見ていくことにしよう。

104～193行のlazyupdate_openからいく。真っ先に，すでに遅延書き込みが有効になっているかどうかを調べ，まだなら内部フラグを“遅延書き込み中”にセットする(109～110行)。続いて，引数として渡されたバッファ領域サイズから何セクタ分のバッファが確保できるか逆算し(113～117行)，そのうえで，バッファ領域とワークを初期化している(120～136行)。各バッファは1024バイトで，それぞれに12バイトの管理情報がつく。便宜上，この管理情報をタグと呼ぼう。通常，このような場面ではバッファ本体とタグはワンセットにしておくものだが，このプログラムでは，バッファ本体は本体で，タグはタグで，配列状にまとめるようになっている。バッファとタグはその並び順によって対応づけられ，先頭から順に使われる。バッファをどこまで使ったかは，タグの配列の使用中部分の直後に負の数を格納することで示し，それとは別に残りバッファ数を示すカウンタが用意されている(多少，冗長ではある)。バッファを使い切ったときには全バッファをフラッシュして，また，頭から使う。一般のディスクキャッシュでは，バッファが足りなくなったら，ひとつだけフラッシュして空きを作るものだが，この場面でそん

幻の遅延書き込みモード

遅延書き込みといえば，Human68k ver.3でサポートされたFASTIO.Xも遅延書き込みモードを備えている。だが，正直なところ僕は，FASTIO.Xの遅延書き込みモードが何なのかを知らない。いろいろ試してみたものの，遅延書き込みモードを意味する-wオプションを指定したときと平時との動作の違いを見つけることはできなかった。いや，一度はわかったつもりだったのだが，それは-s(必ず-wと一緒に指定することになっている)によって連続転送バッファ容量なるものを確保した効果で，-wのではなかったのだ。ただ，-s指定時にはバッファを書き出す順序が変わるのは確かで，その動作は遅延書き込みの効果に通じるものがあることから，ここで取り上げてみたい(実際，これが遅延書き込みモードだったりして？)。

先にFASTIO.Xの位置づけを確認しておこう。一般にはFASTIO.Xはメーカー純正のディスクキャッシュとして認識されているのではないかと思うが，実際にはHuman68k本来のバッファ管理部と置き換わるものであり，その点でふつうの外づけディスクキャッシュとは多少性質が異なる。FASTIO.Xを組み込むとHuman68k内部のバッファは使われなくなり(だから，FASTIO.Xを組み込むときにはCONFIG.SYSのBUFFERSを大きくしてもメモリを無駄にするだけ)，このため，FASTIO.Xのバッファはキャッシュとして以外に，ファイル入出力時のバッファリングにも用いられることになる。

さて，一般にディスクキャッシュでは，バッファが足りなくなったら適当なバッファをひとつ破棄して使い回す。破棄するバッファとしては，最も昔にアクセスされたものを選ぶのがふつうだ。この処理を実現するには，バッファを連結リストの形で管理すると都合がよい。バッファがアクセスされたら必ずリストの先頭に回すようにすると，リストの末尾にはいつでも一番古いバッファがある。また，頻繁に参照されるバッファがリストの前のほうに集まることから，バッファの検索速度の点でもよい効果が得られるだろう。実際，Human68kやFASTIO.Xもこのバッファ管理方式を採用している。だが，この方式は，Human68kにおけるファイル出力のバッファリングと少々相性が悪い。

Human68kでは出力のバッファリング時に，一杯になったバッファをすぐに書き出さず，バッファが足りなくなるか，fflushが発行されるか，ファイルがクローズされるまで溜め込む。一見，これは正しい在り方のように見えるが，バッファ管理アルゴリズムとの絡みで，ちょっと笑える現象を生む。仮にファイルが第20セクタから第23セクタまでの4セクタに書き出されるとして，すべてバッファリングされたとしよう。Human68kは各セクタに順にバッファを割り当てていき，アクセスされたバッファはリストの先頭に移動する。すると，4セクタ分の出力が終わった時点で，リスト上のバッファの並びは，

23→22→21→20

となる。クローズ時などのバッファの掃き出しはリストの順序で行われるから……見事に逆順に書き出されるわけだ。これではディスクの回転待ちがガンガン発生し，事実上，1回転で1セクタしか書き込めない。

FASTIO.Xの-sオプションはこの現象を回避することを目的としているようで，-sを指定すると，連続するセクタへの書き込みの順序は整えられ，かつ，まとめられてデバイスドライバに渡されるようになる。

なことをするのは意味がない。遅延書き込みの効果は，書き込みを溜め，まとめて書き出すことで引き出されるわけで，バッファが一杯になるたびに1セクタずつ書き出していったのでは，語義どおり“本当に書き込みを遅らせるだけ”のジョークプログラムにしかならないだろう。

タグには，そのバッファがどのドライブのどのセクタを保持しているか(各4バイト)に加えて，バッファ本体のアドレス(4バイト)が含まれ，この順序で並ぶ。デバイスドライバレベルではすでにドライブ番号が意味を失っていることから，ドライブの識別には内部DPBの先頭アドレスを使う。なお，バッファ本体とは並び順で対応づけられているというのにわざわざアドレスを覚えておく仕様になっているのは，あとで，シークの最適化を行う際に，タグの配列をソートする都合だ。タグを並べ替えてもバッファとの対応を見失わないよう，タグからバッファアドレスが求められるようになっている。

138～180行で遅延書き込み処理部本体をデバイスドライバに取りつかせる。ここでは，内部DPBを順にたどりつつ，各ドライブに対応するデバイスドライバに対して処理を繰り返している。先頭の内部DPBを得るために，Human68kのワークの未公開部分を参照していたりするが，ほかに手段がないのだからしようがない(141行)。毒を食らわば皿までだ。

ここのループ中では，まず144～149行で，取りつきたくないデバイスドライバを弾いている。前述のように，RAMディスクは外す(注10)。また，このプログラムは1セクタが1024バイトか512バイトのディスクにのみ対応しているので，それ以外も弾く。ちなみに，CD-ROMやネットワークドライバ(実物は見たことがないが)などの特殊なデバイスは，DPB(内外とも)中の“1セクタあたりのバイト数”を0にすることで示されるので，やはり，この時点で弾かれることになる。

151～154行ではデバイスドライバに二重に取りついてしまわないようチェックしている。内部DPBは各ドライブ別々になっているが，ひとつのデバイスドライバが複数のドライブ(デバイスドライバの用語ではユニット)をサポートする場合もあるから，このチェックは欠かせない。ここで，あとの遅延書き込み処理本体ではユニット番号から内部DPBを逆引きする必要があることから，各デバイスドライバごとに26エントリのテーブルを用意し，ここに内部DPBを収めることになっており，157～160行や188～192行でこの処理を行う。テーブル自体はストラテジエントリに引っかけるコードにくっついている(52行)。162～176行でそのコードの複製を行い，デバイスドライバヘッダ中の旧エントリアドレスを待避のうえ，新エントリアドレスに差し替えれば，ひとつのデバイスドライバの征服が完了する。

続いて203～253行のlazyupdate_close。ここでは，内部フラグによって遅延書き込み中かどうかのチェック(211～212行)と，lazyupdate_openを呼び出したプロセスと同じプロセスかどうかのチェック(214～215行)に続き，全バッファをフラッシュして(217行)，先に取りつくために書き換えたデバイスドライバのヘッダを元に戻している(219～227行)。

バッファのフラッシュはほかからも利用するので，260～405行のサブルーチンに抜き出してある。その冒頭263～347行はシークの最適化の処理で，早い話，タグの配列を内部DPBアドレスとセクタ番号をキーにしてソートしているだけだ。ハードディスクの場合，ときにセクタの物理的な位置関係が3次元的になり，本当にヘッドの移動距離を最小にしようとすれば，そのことも考慮する必要があるのだが，その関係はドライブによって異なることから，ここでは，単にセクタ番号の小さい順に書き込むことにしている。ソーティングアルゴリズムは単純挿入法を併用したクイックソートで，配列要素の比較部分は今回の目的に絞ってハードコーディングした。

350～399行で実際にバッファをフラッシュする。とくに触れていなかったが，一旦デバイスドライバから奪って溜めた出力データをディスクに書き込むためには，元のデバイスドライバに頼むしかない。そこで，この部分ではリクエストヘッダを内部的に用意して，Human68kがデバイスドライバを呼び出す手順をエミュレートしている。フラッシュの処理中，363～374行では書き込むセクタ番号が連続していたら，これをなるべくまとめるようにしている。ただし，この件に関してリスト1はあまり積極的ではなく，セクタ番号だけではなくバッファも隣接している場合にしかまとめ込みを行わない。ちゃんとしたプログラムであれば，元々このようなことをしなくても済むように書かれているはずだ，との判断で見切った(注11)。本格的にやりたければ，適当な大きさのメモリを用意して，ここにバッファの内容を転送しつつ大きなブロックにまとめるようにするよう改造することになるが，その場合，メモリ間転送が増えるわけだから，損益分岐点がやや高くなるのも気になるところだ。なお，この連続セクタのまとめ込みについては，FASTIO.Xのほうがよほどうまくやってくれるので，併用するのがいいだろう。

残り2つのサブルーチン239～253行のlazyupdate_flushと413～416行のlazyupdate_calcについては見てのとおりということで，いよいよ，421行からの遅延割り込み処理本体だ。

426～427行でストラテジエントリがセットした情報を取り出したあと，428行では取りついたデバイスドライバの元の割り込みエントリアドレスをスタックの底のほう，待避したレジスタの直後の位置に格納している。このための領域は???行でレジスタを待避するときに，spも一緒にプッシュすることで確保している。ここに積んだアドレスは，あとでデバイスドライバに処理を引き継がせるときに使う。493～494行がその引き継ぎの処理で，レジスタ内容を復帰した時点で，428行で積んだアドレスがスタックトップにくるから，rtsすればそのアドレスに飛べる。

注10）動作試験中は，逆にRAMディスクのみに取りつくようにしたほうが安全かもしれない（145行をbeqからbneにする）。それで暴走したり，RAMディスク上のファイルが壊れたりしないことを確認したら，今度はフロッピーディスクのみを対象に同様の動作試験を行うようにすればさらに安全だろう（145行はbneのまま，144行のRAMIDを$feにする）。

注11）ただし，1セクタ512バイトのデバイスでは連続セクタのまとめ込みがまるで働かない(バッファの前半部のみを使うので，絶対にセクタ内容が隣接しない)のは，完全な手抜き。

また,元のデバイスドライバを経由せずにHuman68kに直接戻るときには，476～478行でやっているように，積んだアドレスを捨ててからrtsする。

430～434行で，リクエストヘッダからコマンドコードを取り出し，遅延書き込み処理部が受け持つコマンドと元のデバイスドライバに任せるコマンドとをふるいにかける。基本的には，セクタの読み込みと書き込み(ベリファイありとなしの両方)を拾い上げて，残りは元のデバイスドライバに任せるのだが，ここでは一時的にドライブコントールも捕まえている。このプログラムはドライブコントロール自体を処理するわけではないが，ディスクのソフト的なイジェクトを検出したいのだ。ドライブのイジェクトボタンが押された場合はどうしようもないとしても，ソフト的なイジェクトであれば，それを捉えて，バッファをフラッシュすることが可能となる。そのイジェクト関係の処理は481～491行で行っている。本来，イジェクトされたドライブについてのみフラッシュすればよいのだが，ここでは無条件に全バッファをフラッシュしてしまっている。なお，491行で，いま処理を横取りしているデバイスドライバのストラテジエントリを呼び出しているのは，バッファのフラッシュの過程で，同じデバイスドライバを呼び出した可能性があるためだ。もし呼び出したのなら，そのデバイスドライバがリクエストヘッダのアドレスを格納していたワークが更新されていることになり，元に戻してやらないと，正しい処理が行えない。そこで，念のため，ストラテジエントリを呼び出して，本来処理すべきだったリクエストヘッダを覚え直させているわけだ。

戻って，440～461行では，セクタの読み書きコマンド共通の前処理を行っている。仮にエラーコードをクリアして(440～442行)，ユニット番号に対応する内部DPBアドレスを取り出し(444～447行)，セクタサイズに応じて１セクタ分のブロック転送を行うサブルーチンを選び(449～454行)，入出力対象となるセクタ番号やセクタ数，および，バッファアドレスを取り出してから(456～457行)，バッファアドレスによっていま一度，ブロック転送ルーチンを選択する(459～461行)という流れだ。ブロック転送ルーチンは598行以下に用意されている。セクタサイズによって1024バイト用と512バイト用があり，それぞれについて，転送対象メモリアドレスが偶数のとき用と奇数のとき用の区別があるので，計４通りに分かれている。

463～464行で読み込みコマンドと書き込みコマンドとで処理を振り分ける。先に515行からの書き込みコマンドの処理を見てもらおう。

516～518行でセクタ数に応じて処理を振り分ける。０なら黙って帰り，１なら521～546行で，２以上なら548～566行で処理する。で，ここがちょっとしたミソなのだが，実はこのプログラムはすべての書き込みをバッファに溜めるのではない。１セクタの書き込みのみ溜めて(521～546行)，複数セクタの書き込みはリアルタイムに処理してしまう。こうすることで，バッファをあまり消費せずに，FATとディレクトリの重複した更新を効果的に減らせる。まとまった複数セクタの単位で書き込まれるのは非常に高い確率でファイル本体であり，ファイル本体が重複して書き込まれることはあまりないことから，溜めてバッファを無駄遣いするよりも素通ししたほうがよい，という考え方だ。なるべく多くの書き込みを溜めたほうがシークの最適化には都合がよいのだが，FATとディレクトリに対する書き込みさえ溜められれば，これらとファイル本体とのあいだのシークが省かれることから，同等の効果は得られる。

ただし，複数セクタの書き込みを完全に素通りさせるわけにはいかない。それらのセクタの中にはバッファに存在するものもあるかもしれないからだ。１セクタだけで書き出したセクタを上書きする形で複数セクタの書き込みを行うことはまず考えられないが，いちおう，検査して，バッファ中にあればバッファを更新しておくようにしておかなければならない(549～564行)。

読み込みコマンドの処理に関しては，セクタ数による処理の振り分けを行う必要はとくになかったのだが，１セクタの場合だけ特別扱いして，多少処理時間を稼いでみた(472～474行)。ここでは，単に，読み込み対象セクタがバッファにあればバッファから転送し，なければ，元のデバイスドライバに任せるという処理を行っている。対して，複数セクタの読み込み部は，もう少し複雑だ。まず元のデバイスドライバの割り込みエントリをサブルーチン的に使って読み込みを行わせ(497～498行)，それから改めて，いま読み込んだセクタがバッファにあったかどうかを調べて，あればバッファ内容で更新する(503～512行)という手順を踏んでいる。場合によっては，二度手間にもなりうるが，先ほどの複数セクタ書き込み同様，確率は低いので，問題にはならないと思う。ただ，指定セクタがバッファ上にあるかどうか調べるサブルーチンsearch(571～585行)は，あまりに馬鹿正直な作りなため，足を引っ張ることもあるかもしれない。

簡易万能ブースター

さて，リスト１とリスト２があれば，ソースがあるあらゆるプログラムに遅延書き込み機能を組み込むことができるようになるわけだが，ソースが必要な以上，Human68k標準コマンドなどのバイナリで提供されるプログラムはその恩恵を受けられない(ディスアセンブルしてもいいけど)。だが，遅延書き込みの効果はメモリ上の全プログラムに伝わるのだから，遅延書き込み環境に入り，子プロセスを起動し，子プロセス終了後，遅延書き込み環境から抜ける，という処理をするだけの小さなプログラムを用意すれば，任意の実行ファイルを遅延書き込み環境で動作させることが可能となる。リスト３のQ.Sがその一例だ。リスト３から実行形式ファイルを作

成するには，

AS Q

LK Q LAZYUPDATE /X

とすればよい。生成されるQ.Xの使い方は次のとおり。

Q コマンドライン

要するに，任意のコマンドラインの頭にプリフィクスの感覚で“Q”を置く。ただし，COMMAND.Xの内部コマンドの場合は，

Q COMMAND DEL A:＊.BAK

のように，“COMMAND”が余分に必要となる。

Q.Xを使えば，多くの既存プログラムが手軽に高速化されるだろう。メーカー純正ツールでいうと，

ATTRIB，CASE，COPYALL，MOVE，TOUCH

のあたりや，COMMAND.X内部コマンドの

さらなる高速ファイルコピーへの道

冒頭で登場した自家製ファイルコピーツールcpには，遅延書き込み以外にもディスクアクセス回数を減らすちょっとした細工が施されており，このため，Q.X＋COPYよりも，もう頭ひとつ速い。といっても，この細工によって節約できるのは1ファイルにつき1セクタ分の読み込みに過ぎず，しかも，それに伴うシークは遅延書き込みによってすでに省略されているがために，通常，さほど大きな速度差は生じない。それでも，読み込みに付随して行われる“ある程度の処理時間を食っていると予想されるHuman68kの内部処理”が一緒に省略できることから，コピーするファイルの数が多ければ，数〜数10秒程度の節約にはなる。遅いマシンほど顕著な効果が見られることを考えると，読み込み自体よりもHuman68kの内部処理を省略するのが効いているらしい。実際，本文で行った4000バイト×192本のコピーテストでは，X68000で61秒，X68030で58秒という結果になり，Q.X＋COPYのX68000で110秒，X68030で76秒という数字と比べると，X68000とX68030の差が一気に縮まっているのがわかる。

で，これからそのトリックを紹介するわけだが，多少ショッキングな話になるから，そのつもりでいておいてほしい。というのも，僕がわざわざ手間をかけて節約した1セクタの読み込みは，本来は最初から存在しないはずのものだからだ。

さて，出力データ中のセクタサイズ以下の端数部分は，Human68k内部でバッファリングされるわけだが，残念なことにHuman68kのバッファリングアルゴリズムには致命的な欠陥があり，日夜，大量の無駄を生み出している。あろうことか，出力データをバッファに転送するのに先立って，そのバッファの掃き出し先となる予定の“空きセクタ”の内容を読んで，バッファを充填しているのだ。

もう少し詳しくいうと，バッファを事前に充填しなければならない場面というものも確かにある。既存ファイルの部分的な更新だ。すでに存在するファイルを書き込みモードでオープンし，先頭1バイトだけを書き換えて，すぐクローズする場合を考えてみよう。もし，この1バイトの出力データを空の(＝ゴミで埋まった)バッファに溜めるようになっていたなら，バッファ内容を1セクタ分ディスクに書き込んだ時点で，バッファに残ったゴミによりファイルの2バイト目以降が上書きされてしまう。だからこの場合には，ファイルの頭の部分をいったん読み込んで，バッファを充填しておかなければならない。一般化すると，ファイルの途中を上書きする際にはバッファの事前充填が必要，となる。

だが，ファイル末尾への追加の場合，バッファにゴミを残したままでも上書きされて困るデータは存在しない。そもそも，空きセクタ内容を読み込んだのでは，ゴミを隠すのに別のゴミを被せるようなもので，まるで意味がない。それなのにHuman68kは，ファイルの途中を上書きしているのか，ファイル末尾に書き足しているのかを区別することなく，無条件にバッファを充填するのだ。

ひょっとすると，Human68kの設計者は“明示的に書き換えられた部分以外はディスクを一切書き換えず，たとえゴミデータであろうとも保存する”という強いポリシーを持っていたのかもしれない(あるいは，MS-DOSもそうなっているとか？)が，お蔭で，Human68kのユーザーは，ファイルを1本コピーするごとに，その末尾の端数部分のバッファリング処理過程で1セクタ分の読み込み(と，それに伴う1回の余分なシーク)の時間をドブに捨てさせられているわけだ。総合計を考えると，ちょっと気が遠くなる。

だが，気を失うのはまだ早い。“ふつうに”ファイルを作成し“ふつうに”データを出力していくと，それは“つねに”ファイル末尾への追加となる点に注目してほしい。ユーザープログラムレベルできちんと出力をバッファリングし，セクタの倍数単位で書き出すようにしていればよいが，中途半端なサイズのデータを無頓着に書き出すと，全出力がHuman68k内部でバッファリングされることになり……これ以上はいわなくてもわかるだろう。

さらに，ユーザープログラムレベルでバッファリングしているからといって，まだ，安心はできない。Human68kは出力データサイズがセクタサイズ“未満”ではなく“以下”のときにバッファリングするため，バッファサイズがセクタサイズちょうどではこの問題から逃れることができないのだ。Human68kでは多くのディスク装置が1セクタ1024バイトとして扱われることを考えると，バッファは最低でも2048バイト用意する必要がある(《入門編》時代にバッファドI/Oを取り上げた際には，まだこのことに気づいていなかったため，バッファサイズを1024バイトにしてしまったことをお詫びしておく)。この点，X68000界で最も広く使われているバッファドI/OルーチンであるXCの標準入出力ライブラリはどうかというと，しっかり1024バイトしか確保していなかったりするわけで，まともにハマっている。

あ，いま慌ててライブラリとヘッダの一部を書き換えてバッファサイズを大きくしようとした人，ちょっと待って。それもあまり意味がない。いわゆるテキストモードでは行終端コードの変換が行われるわけだが，XCのライブラリではこの処理を，

1) バッファから1行切り出して，行終端コードはつけずにDOSコールwriteで書き出す

2) CR＋LFの2バイトをDOSコールwriteで書き出す

というように，行単位で行っている。だから，テキストモードを使う限り，バッファを大きくしても無駄なのだ(合掌)。もちろん，行終端の変換部分も全面的に書き直すというのなら話は別だが。

さて，Human68kのこの仕様に気づいたときには，僕もさすがにがっくりきたが，すぐに気を取り直して，せめてファイルコピー時だけでも，この無駄な空読みが発生しないようにしてやろうと心に決めた。で，考えたわけだ。要は，セクタサイズに満たない端数データさえ存在しなければよい。そうすれば，バッファリング自体が行われないから，空読みも発生しなくなる。もちろん，現実のファイルの大きさはセクタサイズの倍数とは限らないわけだが，足りないのなら，補ってやればよいのだ。

具体的な手順は次のようになる。

1) DOSコールwriteでファイルを書き出す際に，ファイル末尾にセクタサイズ未満の端数があったら，それをセクタサイズの倍数に切り上げて(多少のゴミと一緒に)書き出す

2) クローズする前に，DOSコールseekで本来のファイル末尾にファイルポインタを移動し，書き込みサイズを0にしてwriteを呼び出すことで，ファイルを正しいサイズに切り詰める

予想どおり，これによって1セクタの空読みは姿を消し，前述のように処理時間も短縮された。また，これまでは末尾の1セクタと残りとの2回に分割されていたデバイスドライバの呼び出しが1回にまとめられるようになったため，遅延書き込みバッファの利用効率も上がるという嬉しい副作用もあった。

なお，実装するにあたっては，1セクタ未満のファイルにはこの手法が通用しないことに注意してほしい。この場合，セクタサイズの倍数に切り上げたとしても，セクタサイズの1倍は結局セクタサイズ“以下”だから，バッファリングの対象になってしまう。逆にいうと，無用なオーバーヘッドを避けるため，1セクタ未満のファイルに対しては上記の小細工を行わないようにするべきだ。ただし，試してみたことはないのだが，1クラスタが2セクタ以上のドライブが相手なら，1セクタ未満のファイルを1セクタではなく2セクタ分に水増しするという手が使えるはずだ。1セクタの読み込みを省略するために1セクタ余分に書き込む格好にはなるが，連続するセクタへの書き込みなら1セクタでも2セクタでもたいした時間の差はないし，少なくともバッファリングのソフト的な手間を省くことができるから，トータルの処理時間は短縮されると思う。どうせなら，ここまで試してみてもらいたい。

COPY, DEL, REN

などとは比較的相性がいい。凶悪な使い方としては,

Q MAKE

なんてのも可能だ。

ただし,使用時には,Q.Xの下で実行したプログラムが終了してQ.Xに制御が戻り,さらにQ.X自身が終了するまで,絶対に電源を落としたり,リセットしたり,ディスクをイジェクトしたり(ソフト的に行う場合を除く)してはならないことを頭に入れておいてほしい。対話型のプログラムを実行すると,つい,遅延書き込み中だということを忘れがちだから,

Q COMMAND

とか

Q SXWIN

なんていう使い方はしないほうが身のためだろう。ソフト的なイジェクト発生時にバッファがフラッシュされることを利用して,定期的にCTRL+F1~F4やドライブアイコン上のイジェクトボタンによる手動フラッシュを実行すれば多少安全にはなるが,とても勧められたものではない。

リスト3の内容については特に見るべき点はないが,先ほど触れた,中断時処理アドレスの設定(163~169行)と中断時処理本体(111~123行)はいちおうチェックしておいてほしい。あと,24行の記号定数NBUFFが確保するバッファ数を表しているので,適宜書き換えてみるのもいいだろう。HAS.Xなら,

HAS Q -sNBUFF=100

のようにアセンブル時にコマンドラインから指定することもできる。通常の用途ではデフォルトの32個で十分だとは思うが,ファイルのコピーの場合,ほぼ確実に1本につき1個のバッファが消費される(注12)ことから,バッファ数を大きくすれば,バッファが一杯になって自動的にフラッシュされる回数が減り,多少は速度が改善されるかもしれない(逆に遅くなる場合もあるようだが)。ただし,バッファ数を大きくするにつれて,万一のときに失われるデータ量も増えることに注意してほしい。バッファ数を変えた実行ファイルをいくつか作っておいて,用途に応じて使い分けるのもいいかもしれない。あるいは,オプションで起動時にバッファサイズを指定できるよう改造するのも建設的だ。

注12)ファイル末尾に存在するセクタサイズ未満の部分がHuman68k内部でバッファリングされ,この1セクタの書き出しがSWELLFISH.SYSの網に引っかかるため。

ところで,リスト3では何カ所かに条件つきアセンブルを噛ましてあるのに気づいたと思う。実は,リスト3はBIND.Xにより既存の実行ファイルとバインドして使うこともできるようになっており,シンボルFORBINDを定義してアセンブルすると,このバインド用バージョンが生成される。リスト3の頭の部分に手順をまとめてあるので,参考にしてほしい。なお,この技はすでにバインドされているプログラムには使えないから念のため。

まとめ

最後に,遅延書き込みの効果を示すために簡単なベンチマークテストを行ってみた。表1は,既存のプログラムをふつうに動かした場合と,Q.X上で動かした場合とでの,処理時間の変化を示している。

最初の3つのテストでは,RAMディスク上に作成した4000バイトのファイル192本をCOPYによって空の2HDディスクにコピーし,TOUCH.Xでタイムスタンプを現在日時にし,DELで削除した。このテストはやや作為的なので,4番目のテストでは,多少現実的な例として,COPYALL.Xにより,2HDディスク間でHuman68kシステムディスク中の全ファイルを再帰的にコピーする場合を見ている。残りの4つのテストは条件を変えながらLHA.Xで圧縮されたファイルの展開時間を見たもので,サンプルとしては,ちょうど手元にあった電脳倶楽部Vol.88のIKAP88.LZH(ファイルサイズ約245Kバイト,圧縮率約50%,含まれるファイル数105)を使った。2HDディスク間での展開と,元アーカイブファイルをいったんRAMディスク上に転送してからの展開とを,LHA.XとHLM.R(電脳倶楽部で使われている高速展開ツール)それぞれで試している。

*

今回は,少々大がかりな例ではあったが,遅延書き込みを題材に,ファイルI/Oの最適化を取り上げてみた。個人的には好きな遊びなのだが,受けは悪いかもしれないな,と反省しつつ,また来月(なお,くれぐれも今月のプログラムの利用は慎重に)。

表1 Q.Xの効果(単位:秒)

	X68000(10MHz)	X68030(25MHz)
COPY (RAM→2HD)	233→110	196→ 76
TOUCH (2HD)	33→ 4	32→ 2
DEL (2HD)	48→ 6	48→ 3
COPYALL (2HD→2HD)	162→108	138→ 86
LHA (2HD→2HD)	320→254	243→186
LHA (RAM→2HD)	187→120	113→ 55
HLM (2HD→2HD)	146→ 83	118→ 56
HLM (RAM→2HD)	125→ 61	98→ 43

リスト1 SWELLFISH.S

```
*	遅延書き込みをサポートする常駐ライブラリ
*			Copyright 1995 Toshiyuki Murata
*
*	作成法:
*		as swellfish
*		lk swellfish -oswellfish.sys
*
*	config.sysへの組み込み法:
*		device = swellfish.sys

		.include	doscall.mac

*
*	デバイスドライバヘッダ構造
*
		.offset 0
drvNEXT:	.ds.l	1	*次のデバイスドライバ
drvATR:		.ds.w	1	*デバイス属性
drvSTRENT:	.ds.l	1	*ストラテジエントリ
drvINTENT:	.ds.l	1	*割り込みエントリ

*
*	内部DPB構造
*
		.offset 0
dpbDRIVENO:	.ds.b	1	*起動時のドライブ番号(0...A:)
dpbUNITNO:	.ds.b	1	*ユニット番号
dpbDRIVER:	.ds.l	1	*対応するデバイスドライバ
dpbNEXT:	.ds.l	1	*つぎの内部DPB(-1...最後)
dpbBYTE:	.ds.w	1	*1セクタあたりのバイト数(0...特殊)
```

```
dpbSECT:        .ds.b   1               *1クラスタあたりのセクタ数-1
dpbCLUST2SECT:  .ds.b   1               *(dpbSECT+1)倍するためのシフト数
dpbFATSECT:     .ds.w   1               *FATの先頭セクタ番号
dpbNFAT:        .ds.b   1               *FATの個数
dpbFATNSECT:    .ds.b   1               *FATの占めるセクタ数
dpbROOTNFILE:   .ds.w   1               *ルートの最大ファイル数
dpbDATASECT:    .ds.w   1               *データ部の先頭セクタ番号
dpbMAXFAT:      .ds.w   1               *総クラスタ数+1
dpbROOTSECT:    .ds.w   1               *ルートディレクトリ先頭セクタ番号
dpbMEDIAID:     .ds.b   1               *メディアID
dpbSECT2BYTE:   .ds.b   1               *dpbBYTE倍するためのシフト数
dpbFIRSTFAT:    .ds.w   1               *FAT検索先頭セクタ番号
*
RAMID           equ     $f9             *RAMディスクのメディアID
*
*       ストラテジエントリに引っ掛けるコード
*
                .offset -2
INTJMP:         jmp     0.l                     *元の割り込みエントリへ
ORGINTENT       equ     *-4                     *
DPBTABLE:       .ds.l   26                      *内部DPBのテーブル
HSTRENT:        move.l  a5,reqheader            *新ストラテジエントリ
                move.l  #0,dpbtable
STRJMP:         jmp     0.l                     *元のストラテジエントリへ
ORGSTRENT       equ     *-4                     *
SIZEofSTRHOOK   equ     *+2
*
                .text
                .even
*
*       デバイスドライバヘッダ
*
fugu_header:    .dc.l   -1
                .dc.w   %11000000_00000100      *bit15 ... ｷｬﾗｸﾀﾃﾞﾊﾞｲｽ
                                                *bit14 ... IOCTRL可
                                                *bit2  ... NULデバイス
                .dc.l   fugu_strategyentry
                .dc.l   fugu_interruptentry
                .dc.b   '@FUGU',$20,$20,$20
*
*       ワーク
*
fugu_reqheader: .ds.l   1                       *@FUGUのリクエストヘッダ
reqheader:      .ds.l   1                       *┐リクエストヘッダ
dpbtable:       .ds.l   1                       *|内部DPBのテーブル+α
nbuffer:        .ds.w   1                       *|総バッファ数
counter:        .ds.w   1                       *|空きバッファ数
taglist:        .ds.l   1                       *|バッファ管理情報
buffer:         .ds.l   1                       *|バッファ
strtrap:        .ds.l   1                       *|ｽﾄﾗﾃｼﾞｴﾝﾄﾘを捕捉するコード
ownerpsp:       .ds.l   1                       *┘openしたプロセスのpsp
semaphore:      .ds.b   1                       *
                .even
*
*       内部ルーチンジャンプテーブル
*
                bra.w   lazyupdate_calc         *-4
jumptable:      bra.w   lazyupdate_open         *+0
                bra.w   lazyupdate_flush        *+4
                bra.w   lazyupdate_close        *+8
*
*       遅延書き込み開始
*
*       in      d0 = psp
*               a0 = バッファ領域
*               d1 = バッファ領域バイト数
*       out     d0 = 0 ... 正常終了
*               d0 = 1 ... エラー
*
lazyupdate_open:
SAVREGS         reg     d1/a0-a4
SAVSIZ          set     (1+5)*4
                movem.l SAVREGS,-(sp)
                tas.b   semaphore               *初期化済み?
                bne     openerror               *  そうなら何もしない
                                                *バッファ容量→バッファ個数
                subi.l  #SIZEofSTRHOOK*26+8,d1
                bcs     openerror               *
                divu.w  #1024+12,d1             *d1.w = バッファ個数
                bvs     openerror               *
                beq     openerror               *
                                        *ﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞに取り付く
                lea.l   nbuffer(pc),a2          *バッファを初期化する
                move.w  d1,(a2)+                *nbuffer
                move.w  d1,(a2)+                *counter
                moveq.l #12,d2                  *
                mulu.w  d1,d2                   *d2 = nbuffer*12
                swap.w  d1                      *
                clr.w   d1                      *
                lsr.l   #16-10,d1               *d1 = nbuffer*1024
                add.l   a0,d1                   *d1 = taglist先頭
                move.l  d1,(a2)+                *taglist
                move.l  a0,(a2)+                *buffer
                movea.l d1,a0                   *a0 = taglist
                st.b    (a0)                    *終端マーク
                adda.l  d2,a0                   *
                move.l  jmpinst-2(pc),(a0)+
                move.l  a0,(a2)+                *strtrap
                move.l  d0,(a2)                 *ownerpsp
                lea.l   interruptentry(pc),a3
                                                *a0 = ｽﾄﾗﾃｼﾞｴﾝﾄﾘのフック用領域
                                                *a3 = 割り込みｴﾝﾄﾘのフック先
                move.l  $1c3c.w,d1              *d1 = 最初の内部DPB
dpbloop:        movea.l d1,a1                   *a1 = 内部DPB
                cmpi.b  #RAMID,dpbMEDIAID(a1)       *RAMディスク?
                beq     dpbnext                     *  そうなら飛ばす
                cmpi.w  #1024,dpbBYTE(a1)           *1024バイト/セクタか
                beq     open0                       *  512バイト/セクタ?
                cmpi.w  #512,dpbBYTE(a1)            *
                bne     dpbnext                     *  それ以外は飛ばす
open0:          moveq.l #0,d1
                movea.l dpbDRIVER(a1),a2            *a2 = ﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞ
                move.l  drvINTENT(a2),d0            *d0 = 割り込みｴﾝﾄﾘ
                cmp.l   a3,d0                       *フック済み?
                beq     setdpb                      *  そうならDPBのセットのみ
                move.l  d0,ORGINTENT(a0)    *元の割り込みｴﾝﾄﾘを覚えておく
                move.b  dpbUNITNO(a1),d1    *ユニット番号に対応する
                add.w   d1,d1               *  内部DPBの位置を覚えておく
                add.w   d1,d1               *
                move.l  a1,DPBTABLE(a0,d1.w)
                move.l  a0,d1               *ｽﾄﾗﾃｼﾞｴﾝﾄﾘに引っ掛けるコードを
                lea.l   HSTRENT(a0),a0      *  生成する
                move.l  a0,d0               *
                lea.l   strategyentry(pc),a4
                addq.l  #drvSTRENT,a2       *
                move.l  (a2),ORGSTRENT-HSTRENT(a4)
                move.l  (a4)+,(a0)+         *
                move.l  (a4)+,(a0)+         *
                move.l  d1,(a4)             *
                move.l  (a4)+,(a0)+         *
                move.l  (a4)+,(a0)+         *
                move.l  (a4)+,(a0)+         *
                move.l  (a4)+,(a0)+         *
                movem.l d0/a3,(a2)          *ﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞの両ｴﾝﾄﾘを書き換える
dpbnext:        move.l  dpbNEXT(a1),d1      *すべての内部DPBについて
                bpl     dpbloop             *  繰り返す
                st.b    (a0)                *終端マーク
opendone:       moveq.l #0,d0
openretn:       movem.l (sp)+,SAVREGS
                rts
openerror:      moveq.l #-1,d0
                bra     openretn
setdpb:         move.l  drvSTRENT(a2),a2    *a2 = フック済みのｽﾄﾗﾃｼﾞｴﾝﾄﾘ
                move.b  dpbUNITNO(a1),d1    *ユニット番号に対応する
                add.w   d1,d1               *  内部DPBの位置を覚えておく
                add.w   d1,d1               *
                move.l  a1,DPBTABLE-HSTRENT(a2,d1.w)
                bra     dpbnext
*
*       ストラテジエントリを捕捉するコードのテンプレート
*
strategyentry:  move.l  a5,reqheader        *6  xx xxxx
                move.l  #0,dpbtable         *10 xx 0000 xxxx
                jmp     0.l                 *6  xx 0000
jmpinst:        jmp     0.l                 *2  xx ----
*
*       遅延書き込み終了
*
lazyupdate_close:
SAVREGS         reg     d1-d7/a0-a6
SAVSIZ          set     (7+7)*4
                movem.l SAVREGS,-(sp)
                tst.b   semaphore           *初期化済み?
                beq     closeerror          *
                cmp.l   ownerpsp(pc),d0     *初期化したのと
                bne     closeerror          *  同じプロセス?
                bsr     flushall            *バッファを掃き出す
                movea.l strtrap(pc),a0      *書き換えたﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞﾍｯﾀﾞを
                bra     restorenext         *  元に戻す
restoreloop:    movea.l DPBTABLE(a0),a1     *a1 = ユニット番号0の内部DPB
                movea.l dpbDRIVER(a1),a1    *a1 = 対応するﾃﾞﾊﾞｲｽﾄﾞﾗｲﾊﾞ
                move.l  d0,drvINTENT(a1)    *割り込み/ｽﾄﾗﾃｼﾞｴﾝﾄﾘを復帰
                move.l  ORGSTRENT(a0),drvSTRENT(a1)
                lea.l   SIZEofSTRHOOK(a0),a0
restorenext:    move.l  (a0),d0             *
                bpl     restoreloop         *
                moveq.l #0,d0
                move.b  d0,semaphore
closeretn:      movem.l (sp)+,SAVREGS
                rts
closeerror:     moveq.l #-1,d0
                bra     closeretn
*
*       バッファの掃き出し
*
lazyupdate_flush:
SAVREGS         reg     d1-d7/a0-a6
SAVSIZ          set     (7+7)*4
                movem.l SAVREGS,-(sp)
                tst.b   semaphore           *初期化済み?
                beq     flusherror          *
                bsr     flushall            *バッファを掃き出す
                moveq.l #0,d0
flushretn:      movem.l (sp)+,SAVREGS
                rts
flusherror:     moveq.l #-1,d0
                bra     closeretn
*
*       バッファの掃き出し
*
M               equ     20          *クイックソート→単純挿入法の切り替え点
*
flushall:
                                    *taglistをDPBとセクタ番号でソートする
                                    *(クイックソート+番人付き単純挿入法)
                lea.l   nbuffer(pc),a0      *
                move.w  (a0)+,d7            *d7 = nbuffer
                sub.w   (a0)+,d7            *d7 = nbuffer-counter
                movea.l (a0),a0             *a0 = taglist
                mulu.w  #12,d7              *
                lea.l   0(a0,d7.l),a3       *a3 = taglist末尾
                moveq.l #1,d6               *
                lea.l   M*12.w,a5           *
                                            *
                clr.l   -(sp)               *
                bra     sortloop0ent        *
                                            *
sortretry:      movea.l d0,a0               *
                movea.l (sp)+,a3            *
                                            *
sortloop0:      move.l  a3,d7               *
                sub.l   a0,d7               *
sortloop0ent:   cmp.l   a5,d7               *
```

```
                bcs     isort           *
                lsr.l   #1,d7           *
                btst.l  d6,d7           *
                beq     split           *
                subq.l  #6,d7           *
split:          lea.l   0(a0,d7.l),a4   *
                movem.l (a4),d0-d2      *
                movea.l a0,a1           *
                movea.l a3,a2           *
                bra     sortloop1ent    *
                                        *
sortloop1:      lea.l   -12(a1),a1      *
                move.l  (a2)+,(a1)+     *
                move.l  (a2)+,(a1)+     *
                move.l  (a2)+,(a1)+     *
                movem.l d3-d5,-(a2)     *
sortloop1ent:                           *
sortloop2:      subq.l  #8,a2           *
                move.l  (a2),d4         *
                cmp.l   -(a2),d0        *
                bcs     sortloop2       *
                bne     sortloop3       *
                cmp.l   d4,d1           *
                bcs     sortloop2       *
sortloop3:      movem.l (a1)+,d3-d5     *
                cmp.l   d3,d0           *
                bhi     sortloop3       *
                bne     sortnext1       *
                cmp.l   d4,d1           *
                bhi     sortloop3       *
                                        *
sortnext1:      cmpa.l  a1,a2           *
                bcc     sortloop1       *
                lea.l   -12(a1),a1      *
                                        *
                cmpa.l  a4,a1           *
                bcc     pushleft        *
                                        *
pushright:      movem.l a1/a3,-(sp)     *
                movea.l a1,a3           *
                bra     sortloop0       *
                                        *
pushleft:       movem.l a0/a1,-(sp)     *
                movea.l a1,a0           *
                bra     sortloop0       *
                                        *
isort:          subi.w  #12*2,d7        *┐ 単純挿入法
                bcs     sortnext0       *|
                divu.w  #12,d7          *|
                lea.l   -12(a3),a1      *|
isortloop1:     movea.l a1,a2           *|
                move.l  -(a1),d2        *|
                move.l  -(a1),d1        *|
                move.l  -(a1),d0        *|
                bra     isortloop2ent   *|
isortloop2:     movem.l d3-d5,-12*2(a2) *|
isortloop2ent:  movem.l (a2)+,d3-d5     *|
                cmp.l   d3,d0           *|
                bhi     isortloop2      *|
                bne     insert          *|
                cmp.l   d4,d1           *|
                bhi     isortloop2      *|
insert:         movem.l d0-d2,-12*2(a2) *|
isortnext1:     dbra    d7,isortloop1   *┘
                                        *
sortnext0:      move.l  (sp)+,d0        *
                bne     sortretry       *

                                *バッファを掃き出す
                lea.l   -32(sp),sp      *(
                movea.l sp,a5           *a5 = リクエストヘッダ
                move.l  #$1a_00_08_00,(a5)
                movea.l taglist(pc),a0  *
                bra     flushnext       *
flushloop:      movea.l d0,a3           *a3 = 内部DPB
                moveq.l #1,d1           *d1 = まとめて書き出すセクタ数
                move.l  (a0)+,d2        *d2 = 書き出す先頭セクタ番号
                move.l  d2,22(a5)       *
                movea.l (a0)+,a4        *a4 = バッファ先頭
                move.l  a4,14(a5)       *
                move.w  dpbBYTE(a3),d3  *d3 = バイト/セクタ

                bra     mergenext       *連続する複数セクタへの書き込みを
mergeloop:      addq.l  #1,d2           *  なるべくまとめる
                adda.w  d3,a4           *
                cmp.l   4(a0),d2        *
                bne     merged          *
                cmpa.l  8(a0),a4        *
                bne     merged          *
                addq.l  #1,d1           *
                lea.l   12(a0),a0       *
mergenext:      cmpa.l  (a0),a3         *
                beq     mergeloop       *
merged:         move.l  d1,18(a5)       *まとめて書き出すセクタ数

                move.b  dpbUNITNO(a3),1(a5)     *ユニット番号
                move.b  dpbMEDIAID(a3),13(a5)   *メディアID
                movea.l dpbDRIVER(a3),a2        *a2 = デバイスドライバ

                movea.l drvSTRENT(a2),a2
flushretry:     jsr     (a2)            *ストラテジエントリを呼び出す
                movea.l dpbtable(pc),a1 *
                jsr     INTJMP(a1)      *割り込みエントリを呼び出す
                move.b  3(a5),d0        *エラー?
                beq     flushnext

                                *エラー発生
                move.w  4(a5),d7        *
                move.b  d0,d7           *d7.w = エラーコード
                andi.w  #.not.$1000,d7  *「中止」はできなくする
                cmpi.w  #$0100,d7       *その結果選択枝がなくなったら
                bcs     flushnext       *  無視する
                trap    #14             *エラーハンドラを呼び出す
                subq.w  #1,d7           *再実行?
                beq     flushretry      *  そうならリトライ

flushnext:      move.l  (a0)+,d0        *全バッファに対して
                bpl     flushloop       *  繰り返す
                lea.l   32(sp),sp       *)

                lea.l   nbuffer(pc),a0  *バッファを再初期化する
                move.w  (a0)+,(a0)+     *counter = nbuffer
                movea.l (a0),a0         *a0 = taglist
                st.b    (a0)            *終端マーク
                rts

*
*       バッファ領域に必要なメモリサイズを求める
*
*       in      d0.w = バッファ数
*       out     d0 = 必要なバイト数
*
lazyupdate_calc:
                mulu.w  #1024+12,d0
                addi.l  #SIZEofSTRHOOK*26+8,d0
                rts

*
*       遅延書き込み処理本体
*
interruptentry:
SAVREGS         reg     d0-d7/a0-a6
SAVSIZ          set     (8+7)*4
                movem.l SAVREGS/sp,-(sp)  *sp はダミー

                movea.l reqheader(pc),a5  *a5 = リクエストヘッダ
                movea.l dpbtable(pc),a6   *a6 = DPBテーブル
                move.l  ORGINTENT(a6),SAVSIZ(sp)

                move.b  2(a5),d0          *d0 = I/Oリクエストコマンド
*                         5432109876543210
                move.w  #%0000001100110000,d1
                btst.l  d0,d1             *セクタの読み(4)書き(8,9)と
                beq     chain             *  ドライブコントロール(5)以外は
                                          *  デバイスドライバ本体に任せる

                subq.b  #5,d0             *ドライブコントロール?
                beq     drvctrl           *

                moveq.l #0,d1             *仮にエラーなし
                move.b  d1,3(a5)          *
                move.b  d1,4(a5)          *

                move.b  1(a5),d1          *d1.w = ユニット番号
                add.w   d1,d1             *
                add.w   d1,d1             *
                movea.l DPBTABLE(a6,d1.w),a2    *a2 = d1に対応する内部DPB

                lea.l   longcopy1024(pc),a3
                cmpi.w  #1024,dpbBYTE(a2)       *1024バイト/セクタ?
                beq     readwrite
                cmpi.w  #512,dpbBYTE(a2)        *512バイト/セクタ?
                bne     chain
                lea.l   longcopy512(pc),a3

readwrite:      movem.l 14(a5),d1-d3      *d1 = a6 = 入力バッファor出力データ
                movea.l d1,a6             *d2 = セクタ数
                                          *d3 = セクタ番号
                andi.b  #1,d1             *バッファが奇数アドレスなら
                beq     switch            *  バイト単位転送に切り替える
                movea.l -(a3),a3          *

switch:         addq.b  #5-4,d0           *入力?
                bne     write             *  違うなら出力

read:                           *セクタ読み込み
                subq.l  #1,d2             *dbraを考慮
                bcs     chain             *セクタ数が0だった
                bne     multiread         *セクタ数が2以上だった

                                *1セクタの読み込み
                bsr     search            *該当セクタをバッファから探す
                bne     chain             *見つからなかった
                jsr     (a3)              *1セクタ分転送する

intrretn:       movem.l (sp)+,SAVREGS     *真っ直ぐ呼び出し元に帰る
                addq.l  #4,sp
                rts
*
drvctrl:                        *ドライブコントロール
                cmpi.b  #1,13(a5)         *イジェクト?
                bne     chain             *

                                *イジェクトされようとしている
                move.l  a6,-(sp)          *バッファを掃き出す
                move.l  a5,-(sp)          *
                bsr     flushall          *
                movea.l (sp)+,a5          *
                movea.l (sp)+,a6          *

                jsr     STRJMP(a6)        *元のストラテジエントリを呼び出す

chain:          movem.l (sp)+,SAVREGS     *元の割り込みエントリへ
                rts                       *(sp) = 元の割り込みエントリ
*
multiread:                      *複数セクタの読み込み
                movea.l SAVSIZ(sp),a0     *とりあえず
                jsr     (a0)              *  オリジナルの割り込みエントリを呼ぶ
                tst.b   3(a5)             *エラーなら
                beq     readloopl         *
                bra     intrretn          *  真っ直ぐ帰る

readlooph:      swap.w  d2                *今読んだセクタが
readloopl:      bsr     search            *  バッファにもあれば
                bne     readnext          *  バッファを更新する
                jsr     (a3)              *
                bra     readnext0         *
readnext:       adda.w  dpbBYTE(a2),a6    *
readnext0:      addq.l  #1,d3             *
                dbra    d2,readloopl      *
                swap.w  d2                *
                dbra    d2,readlooph      *
                bra     intrretn
*
write:                          *セクタ書き込み
                subq.l  #1,d2             *dbra を考慮
                bcs     chain             *セクタ数が0だった
                bne     multiwrite        *セクタ数が2以上だった

                                *1セクタの書き込み
                bsr     search            *該当セクタをバッファから探す
                beq     hit               *見つかった

                subq.w  #1,counter        *バッファに空きはある?
                bcc     new               *  ある

                                *バッファが一杯
                movem.l d3/a2/a3/a6,-(sp) *バッファを掃き出す
                bsr     flushall          *
                movem.l (sp)+,d3/a2/a3/a6 *

                lea.l   counter(pc),a1
```

▶やっとXCを購入しました。たった数十行のプログラムでADPCMが操作できてしまうことに驚かされました。標準でADPCMがついているとはいえ，学校で使っているPC-98のTURBO Cと比べると，プログラミングする楽しさが格段に違います。 弧塚 一浩(22)栃木県

```
                subq.w   #1,(a1)+        *counter = nbuffer-1
                movea.l  (a1)+,a0        *a0 = taglist
                movea.l  (a1),a1         *a1 = buffer
                addq.l   #4,a0           *

                                *新規登録
new:            move.l   a2,-4(a0)       *内部DPB
                move.l   d3,(a0)+        *セクタ番号
                move.l   a1,(a0)+        *バッファ
                st.b     (a0)            *終端マーク

hit:            exg.l    a1,a6           *ディスクに書き込まれる筈のデータを
                jsr      (a3)            *  バッファに溜めて
                bra      intrretn        *  真っ直ぐ帰る
*
multiwrite:                     *複数セクタの書き込み
                clr.w    -(sp)           *書き込むセクタが
                swap.w   d2              *  バッファにあれば
writelooph:     swap.w   d2              *  バッファを更新する
writeloopl:     bsr      search          *
                bne      writenext       *
                exg.l    a1,a6           *
                jsr      (a3)            *
                exg.l    a1,a6           *
                bra      writenext0      *
writenext:      adda.w   dpbBYTE(a2),a6  *
                st.b     (sp)            *
writenext0:     addq.l   #1,d3           *
                dbra     d2,writeloopl   *
                swap.w   d2              *
                dbra     d2,writelooph   *
                move.w   (sp)+,d0        *バッファにないセクタがあったら
                bne      chain           *  元の割り込みｴﾝﾄﾘへ
                bra      intrretn        *全セクタキャッシュできたのなら
                                         *  真っ直ぐ帰る
*
*       指定のセクタがバッファ中にあるかどうか調べる
*
search:
                movea.l  taglist(pc),a0
                movea.l  buffer(pc),a1
                move.l   (a0)+,d0
                bmi      searchretn      *Z=0, N=1
searchloop:     cmpa.l   d0,a2           *内部DPBが一致する?
                bne      searchnext
                cmp.l    (a0),d3         *セクタ番号が一致する?
                beq      searchretn      *Z=1
searchnext:     addq.l   #8,a0
                lea.l    1024(a1),a1
                move.l   (a0)+,d0
                bpl      searchloop
                                         *Z=0, N=1
searchretn:     rts

*
*       52バイトブロック転送マクロ
*
COPY52          macro    SOUR,DEST,NTH
                movem.l  (SOUR)+,d0-d7/a0/a2-a5
                movem.l  d0-d7/a0/a2-a5,52*NTH(DEST)
                .endm

*
*       1セクタ分のブロック転送
*
bytecopy1024:   moveq.l  #1024/8-1,d0
bytecopyloop:   move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                move.b   (a1)+,(a6)+
                dbra     d0,bytecopyloop
                rts
*
bytecopy512:    moveq.l  #512/8-1,d0
                bra      bytecopyloop
*
                .dc.l    bytecopy1024
longcopy1024:   movem.l  d2/d3/a2,-(sp)
                COPY52   a1,a6,0
                COPY52   a1,a6,1
                COPY52   a1,a6,2
                COPY52   a1,a6,3
                COPY52   a1,a6,4
                COPY52   a1,a6,5
                COPY52   a1,a6,6
                COPY52   a1,a6,7
                COPY52   a1,a6,8
                COPY52   a1,a6,9
                COPY52   a1,a6,10
                COPY52   a1,a6,11
                COPY52   a1,a6,12
                COPY52   a1,a6,13
                COPY52   a1,a6,14
                COPY52   a1,a6,15
                COPY52   a1,a6,16
                COPY52   a1,a6,17
                COPY52   a1,a6,18
                lea.l    1024-4(a6),a6
                movem.l  (a1)+,d0-d7/a0
                movem.l  d0-d7,-4*8(a6)
                move.l   a0,(a6)+
                movem.l  (sp)+,d2/d3/a2
                rts
*
                .dc.l    bytecopy512
longcopy512:    movem.l  d2/d3/a2,-(sp)
                COPY52   a1,a6,0
                COPY52   a1,a6,1
                COPY52   a1,a6,2
                COPY52   a1,a6,3
                COPY52   a1,a6,4
                COPY52   a1,a6,5
                COPY52   a1,a6,6
                COPY52   a1,a6,7
                COPY52   a1,a6,8
                lea.l    512-4(a6),a6
                movem.l  (a1)+,d0-d7/a2-a4
                movem.l  d0-d7/a2-a3,-4*10(a6)
                move.l   a4,(a6)+
                movem.l  (sp)+,d2/d3/a2
                rts

*
*       デバイスドライバ部
*
fugu_strategyentry:
                move.l   a5,fugu_reqheader
                rts
*
fugu_interruptentry:
SAVREGS         reg      d0/a0/a5
                movem.l  SAVREGS,-(sp)

                movea.l  fugu_reqheader(pc),a5
                move.b   2(a5),d0        *初期化?
                beq      fuguinit        *
                subq.b   #3,d0           *IOCTRLによる入力?
                bne      fuguerror       *

                movem.l  14(a5),d0/a0    *d0 = バッファ
                                         *a0 = 入力要求サイズ
                btst.l   #0,d0           *バッファアドレスは偶数?
                bne      fuguerror       *
                exg.l    d0,a0           *a0 = バッファ
                                         *d0 = 入力要求サイズ
                subq.l   #4,d0           *入力要求サイズは4バイト?
                bne      fugudone        *
                move.l   #jumptable,(a0) *  そうならｼﾞｬﾝﾌﾟﾃｰﾌﾞﾙを返す

fugudone:       sf.b     3(a5)           *正常終了
                sf.b     4(a5)           *
fuguretn:       movem.l  (sp)+,SAVREGS
                rts

fuguerror:      move.b   #$03,3(a5)      *無効なコマンド
                move.b   #$50,4(a5)      *
                bra      fuguretn
*
fuguinit:       pea.l    title(pc)       *初期化
                DOS      _PRINT
                addq.l   #4,sp

                move.l   #fuguinit,14(a5)
                bra      fugudone
*
CR      equ     $0d
LF      equ     $0a
*
title:          .dc.b    CR,LF
                .dc.b    'Swellfish Oh!X Nov.1995 version',CR,LF
                .dc.b    0

                .end
```

リスト2 LAZYUPDATE.S

```
*       Swellfish.sys とのインタフェイス
*                       Copyright 1995 Toshiyuki Murata
*
*       int     lazyupdate_open(size_t *mem, size_t memsiz);
*       int     lazyupdate_flush(void);
*       int     lazyupdate_close(void);
*       size_t  lazyupdate_calc(short nbuff);
*       int     lazyupdate_exist(void);

                .include        doscall.mac

                .xdef   lazyupdate_open
                .xdef   lazyupdate_flush
                .xdef   lazyupdate_close
                .xdef   lazyupdate_calc
                .xdef   lazyupdate_exist
                .xdef   _lazyupdate_open        *for C
                .xdef   _lazyupdate_flush       *
                .xdef   _lazyupdate_close       *
                .xdef   _lazyupdate_calc        *
                .xdef   _lazyupdate_exist       *
*
OPEN            equ     0
FLUSH           equ     4
CLOSE           equ     8
CALC            equ     -4
*
CALL            macro   CALLNO
        .if     CALLNO.slt.0
                subq.l  #-CALLNO,(sp)
        .elseif CALLNO.sgt.0
                addq.l  #CALLNO,(sp)
        .endif
                DOS     _SUPER_JSR
                .endm
*
                .offset 4               *lazyupdate_openの引数構造
MEM:            .ds.l   1               *バッファ用メモリ領域
MEMSIZ:         .ds.l   1               *同バイト数
*
                .offset 4               *lazyupdate_calcの引数構造
                .ds.w   1
NBUFF:          .ds.w   1               *バッファ数
*
                .text
                .even
*
jumptable:      .dc.l   -1
*
lazyupdate_open:
_lazyupdate_open:
SAVREGS         reg     d1/a0
SAVSIZ          set     (1+1)*4
                movem.l SAVREGS,-(sp)
                move.l  jumptable(pc),d0
                bpl     open
                bsr     getjumptable
                bmi     openretn
open:           move.l  d0,-(sp)
                movea.l MEM+SAVSIZ+4(sp),a0
                move.l  MEMSIZ+SAVSIZ+4(sp),d1
                DOS     _GETPDB
```

```
                CALL    OPEN
                addq.l  #4,sp
openretn:       movem.l (sp)+,SAVREGS
                rts
*
lazyupdate_flush:
_lazyupdate_flush:
                move.l  jumptable(pc),d0
                bpl     flush
                bsr     getjumptable
                bmi     flushretn
flush:          move.l  d0,-(sp)
                CALL    FLUSH
                addq.l  #4,sp
flushretn:      rts
*
lazyupdate_close:
_lazyupdate_close:
                move.l  jumptable(pc),-(sp)
                bmi     closeretn
                DOS     _GETPDB
                CALL    CLOSE
closeretn:      addq.l  #4,sp
                rts
*
lazyupdate_calc:
_lazyupdate_calc:
                move.l  jumptable(pc),d0
                bpl     calc
                bsr     getjumptable
                bmi     calcerror
calc:           move.l  d0,-(sp)
                move.w  NBUFF+4(sp),d0
                CALL    CALC
                addq.l  #4,sp
                rts
calcerror:      moveq.l #0,d0
                rts
*
lazyupdate_exist:
_lazyupdate_exist:
getjumptable:
                move.l  d1,-(sp)        *{

                moveq.l #0,d1           *@fugu をオープンする
                move.w  d1,-(sp)        *
                pea.l   devname(pc)     *
                DOS     _OPEN           *
                move.w  d0,d1           *オープンできなければ
                bmi     getretn         *  swellfish.sys がいない

                move.l  d1,(sp)         *デバイス属性を取得する
                DOS     _IOCTRL         *
                andi.w  #$4084,d0       *IOCTRL可の
                cmpi.w  #$4084,d0       *  NULキャラクタデバイスでなければ
                bne     getclose        *  swellfish.sys ではない

                pea.l   4.w             *ジャンプテーブルの
                pea.l   jumptable(pc)   *  ベースアドレスを得る
                move.l  d1,-(sp)        *
                addq.w  #2,(sp)         *
                DOS     _IOCTRL         *
                lea.l   12(sp),sp       *

getclose:       move.w  d1,(sp)
                DOS     _CLOSE
getretn:        addq.l  #6,sp
                movem.l (sp)+,d1        *}
                move.l  jumptable(pc),d0
                rts
*
devname:        .dc.b   '@fugu',0

                .end
```

リスト3 Q.S

```
*       Swellfish.sys の遅延書き込み環境下で任意のコマンドを実行する
*                       Copyright 1995 Toshiyuki Murata
*
*       作成法:
*       <スタンドアロンバージョン>
*               as q
*               as lazyupdate
*               lk -x q lazyupdate
*
*       <バインド用バージョン>
*               as q -sFORBIND -oq2.o
*               as lazyupdate
*               ren foo.x foo.org   (fooは任意の.xファイル)
*               lk -x q2 lazyupdate -ofoo.x
*               bind foo foo.org

                .include        doscall.mac
*
                .xref   lazyupdate_open
                .xref   lazyupdate_close
                .xref   lazyupdate_calc
*
.ifndef NBUFF
NBUFF           equ     32      *確保するセクタバッファ数
.endif
.fail   NBUFF.slt.1
.fail   NBUFF.sgt.32767
*
STRCPY          macro   sptr,dptr
                local   loop
loop:           move.b  (sptr)+,(dptr)+
                bne     loop
                .endm
*
                .text
                .even
*
entry:
                movea.l a1,sp

                DOS     _VERNUM         *Human68kの
                cmpi.w  #$0200,d0       *  バージョンチェック
                bcs     versionerror    *

                pea.l   NBUFF.w         *バッファ領域サイズを得る
                bsr     lazyupdate_calc *
*               addq.l  #4,sp           *
                move.l  d0,d7           *d7 = バッファ領域サイズ
                lea.l   0(a1,d7.l),a3   *a1 = バッファ領域先頭
                                        *a3 = バッファ領域末尾

                lea.l   16(a0),a0       *バッファ用メモリを確保しつつ
                suba.l  a0,a3           *  子プロセス用に
                movem.l a0/a3,-(sp)     *  メモリを解放する
                DOS     _SETBLOCK       *
*               addq.l  #8,sp           *
                tst.l   d0              *
                bmi     nomemerror      *
.ifndef FORBIND
                tst.b   (a2)+           *引数がなければヘルプ
                beq     usage           *
.endif
                pea.l   atbreak(pc)     *中断時の戻りアドレスを設定
                move.w  #_CTRLVC,-(sp)  *
                DOS     _INTVCS         *
                addq.l  #aterror-atbreak,2(sp)
                addq.w  #_ERRJVC-_CTRLVC,(sp)
                DOS     _INTVCS         *
*               addq.l  #6,sp           *

                lea.l   -512(sp),sp     *コマンドライン展開用メモリを確保し
                movea.l sp,a4           *  コマンドラインをそこにコピー
.ifdef  FORBIND                         *
                lea.l   $80-16(a0),a3   *
                STRCPY  a3,a4           *
                subq.l  #1,a4           *
                lea.l   $c4-16(a0),a3   *
                STRCPY  a3,a4           *
                move.b  #$20,-1(a4)     *
                addq.l  #1,a2           *
.endif                                  *
                STRCPY  a2,a4           *

                clr.l   -(sp)           *コマンド名と引数の分離と
                pea.l   256+4(sp)       *  pathからの検索
                pea.l   8(sp)           *
                move.w  #2,-(sp)        *
                DOS     _EXEC           *
                tst.l   d0              *
                bmi     childerror      *

                move.l  d7,-(sp)        *遅延書き込みon
                pea.l   (a1)            *
                bsr     lazyupdate_open *
                addq.l  #8,sp           *
.ifndef FORBIND
                clr.w   (sp)            *子プロセスを起動
.else                                   *
                move.w  #$0100,(sp)     *
.endif                                  *
                DOS     _EXEC           *
*               lea.l   14(sp),sp       *
                move.l  d0,(sp)         *戻り値を待避
                bsr     lazyupdate_close  *遅延書き込みoff

                tst.w   (sp)+
                bmi     childerror
                DOS     _EXIT2          *子プロセスの終了コードを
                                        *  そのまま返して終了
*
atbreak:                        *CTRL+Cによる中断時の処理
                move.w  #$0200,-(sp)
                bra     abort
aterror:                        *ハードウェアエラーによる中断時の処理
                move.w  #$03fc,-(sp)
abort:          DOS     _GETPDB         *pspを取得する
                lea.l   entry-$f0(pc),a0
                cmp.l   a0,d0           *自身のpspと比較する
                bne     quickabort      *不一致なら単に終了する

                bsr     lazyupdate_close  *遅延書き込みoff

quickabort:     DOS     _EXIT2
*
STDERR          equ     2
LF              equ     $0a
CR              equ     $0d
*
.ifndef FORBIND
usage:          lea.l   usagemes(pc),a0
                bra     error
.endif
versionerror:   lea.l   versionerrmes(pc),a0
                bra     error
nomemerror:     lea.l   nomemerrmes(pc),a0
                bra     error
childerror:     lea.l   childerrmes(pc),a0
error:          move.w  #STDERR,-(sp)
                pea.l   (a0)
                DOS     _FPUTS
*               addq.l  #6,sp
                move.w  #-1,-(sp)
                DOS     _EXIT2
*
versionerrmes:  .dc.b   '実行にはHuman68k Ver.2.0以上が必要です',CR,LF,0
nomemerrmes:    .dc.b   'メモリが足りません',CR,LF,0
.ifdef  FORBIND
childerrmes:    .dc.b   'メインモジュールが起動できません',CR,LF,0
.else
childerrmes:    .dc.b   '指定のコマンドが起動できません',CR,LF,0
usagemes:       .dc.b   'Q.X  Oh!X Nov.1995 version',CR,LF
                .dc.b   '機 能: 遅延書き込み環境下でのコマンド実行',CR,LF
                .dc.b   '使用法: q <コマンド>',CR,LF
                .dc.b   0
.endif
*
                .stack
                .even
*
                .ds.l   2048
inisp:
                .end    entry
```

THE SENTINEL

〈対応機種一覧〉 ●MZ-80 K/C/700/1500 ●MZ-80 B/2000 ●MZ-2500/2861 ●X1 ●X1 turbo/Z ●PC-8001/8801/88 ●SMC-777/C ●PASOPIA/5 ●PASOPIA/7 ●FM-7/77/AV ●MSX/2/2+/turbo R ●PC-286/386/486/9801/98/9821 ●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第163部　PICT Puzzle

●PICTパズルだあ

10月号の予告どおり，今月はS-OS"SWORD" 用の「PICT Puzzle」が登場です。画面サイズの関係上，16×16までのサイズしかサポートされていませんが，お手軽練習用と考えると結構遊べるものです。

以前，無責任に「簡単にできそうじゃないですか」と書きましたが，やはり画面サイズはかなりのネックになったようです(ほかに問題となりそうなことはありませんか)。ヒント数字に16進を使うということからも，S-OSでいかに実現するかという苦労の一面が見られます。

X68000版のときもそうだったのですが，描画エリアの半分のサイズを，ヒントのためのエリアとして用意しなければならないため，画面デザインは結構しんどいものです。確かにヒントエリアを最大に使うようなパズルは，めったに作りませんし，作ったとしてもパズルとしてはかなり難しくなってしまいます。ある程度PICTパズルで遊ぶとわかりますが，ヒントとしての1が多くなると，ただ単に難しくなるだけで，面白くなるとはかぎらないのです。

しかし，ヒントエリアを限定させるということは，問題作成の自由度も失うことにもなります。それでは自由に問題を作ることができなくなり，面白さが半減してしまうことにもなりかねません。S-OS版は，小さいけれどPICTパズルの面白さは味わえますので，楽しく遊んでくださいね。

なお，掲載するにあたって5桁ごとのマーク "+" 表示とヒントエリアの表示，問題の内容を多少修正させていただきました。

●S-OS "SWORD" MOOK年内完成!?

配布時期未定のまま，ずるずるとここまできてしまいましたが，年内完成を目標として，予定どおり(?)希望者に実費で配布する形態を取ることが決定しました。

現段階で決まっている内容は，

・全128ページのマニュアル(ゲーム関係の記事はすべて5"2HDにオンラインマニュアルの形にして収録する予定)

・アプリケーションは，5"2HDと5"2D×2に詰め込めるだけ詰め込む(5"2HDと5"2Dの内容は同じもの)

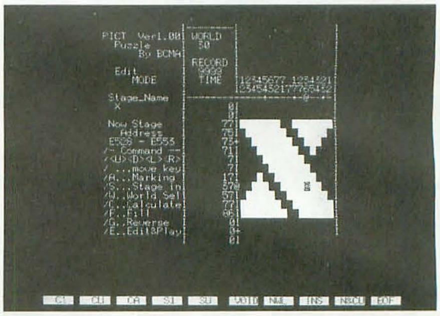
・収録されるアプリケーションはフリーソフト化されたものを中心とする(フリーソフト化されたものは，ほぼすべて収録されると考えて結構です)

・MOOKの価格は，マニュアル+FD(5"2HD，5"2D×2のセット)で送料込み3,000円以内の予定

・S-OS "SWORD" システム本体は含まれません

・マニュアルは完全コピーフリー。プログラムについては，フリーソフト化されたものについては自由にコピーしてかまいません

以上のとおりです。今月号のアンケートハ



ガキにあるとおり，これによって最終的な部数が決定されます。それと同時に購入の意志ありと判断させていただき，自動的に購入者リストに登録されます。ちなみに購入希望者が少ないからといって制作が中止されることはありません。

購入希望者には，MOOKの完成2週間前に完成版の内容と，支払いの手段を明記した申込書を送付します。そして，支払いを確認後，S-OS"SWORD"MOOKを配布することになります。気が変わって購入する意志がなくなったのであれば，そのまま支払いをせずにいてください。それでは，よろしくお願いします。

●CUBEの修正

最後に10月号で掲載したSLANG用パズルゲーム「CUBE」のエディタ部分に不都合が見つかってしまいました。リスト1の関数seselect()内の885～886行の間に，以下のリストを挿入して修正を加えておいてください。

```
if ((edx > maxx) or ((edx == maxx)
and ((edy > lmaxy))) [
    edx = maxx;
    edy = lmaxy;
    prtlist();
]
```

1995■インデックス

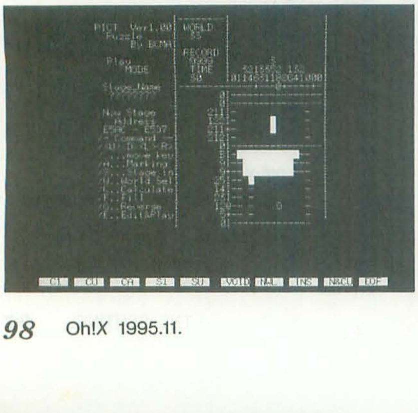
全機種共通
S-OS"SWORD"要

PICT Puzzle

(要SLANG)

Morisawa Miyu
森沢 美優

S-OSにも「PICTパズル」が登場です。画面クリアの関係上，16×16マスまでしか表示できませんが，ゲームの面白さを十分に味わえる作品です。S-OSユーザーの皆さん、元気にハマりましょう。



巻で大人気(?)の「PICTパズル」が，とうとうS-OSに登場です。最近私はSLANGに凝っているので，SLANG用です。ただし，画面の大きさの都合上，16×16までしか表示できません。がんばればなんとかなりますが，視認性が悪くなるのでやめました。これだけの大きさとはいえ，結構遊べるでしょう。

入力方法

まずリスト2のWORLD DATAをMACINTO-Cなどのツールを使って入力してください。チェックサムを確認後，

#S PICDATA.DAT:E000:F15C:E000

以上のように，セーブしておきましょう。

そして，メインプログラムです。

適当なエディタを使い，リスト1を打ち込んでください。あっと，行番号はいりませんよ。そしたら，"PICTpuzzle.SLG"でセーブします(別になんでもいいけど)。そして，SLANGを立ち上げて，

]C PICTpuzzle.SLG

以上のようにコンパイルしましょう。コンパイル後，ERRORが0になったら，一度コンパイラから抜け，コマンドモードに戻り，

#S PICTpuzzle.OBJ:B000:xxxx:B000

としてセーブしましょう(xxxxはプログラムの最終アドレスです)。以後コンパイルする必要はなくなります。

無事コンパイルできたら，

#L PICTpuzzle.OBJ

で，プログラムをロードします。次に，

#L PICDATA.DAT

でWORLD DATAをロードし，

#JB000

でゲーム開始です。なお，おまけのデータファイル初期化プログラム(リスト3)も同様に，

]CPICT_CLEAR.SLG

でコンパイルし，

#S PICT_CLEAR.OBJ:B000:B499:B000

としてセーブしておきましょう。

そして，

#JB000

で，プログラムが実行され，データが初期化されます。

ルールと遊び方

このゲームのルールは，

1) マス目の外にある数字は，その列の中で連続して塗りつぶすマス目の数を表しています

2) 複数の数字が並んでいる場合は，上や左から順に指示されたマス目を塗りつぶしていきます

3) 数字と数字の間は1マス以上の白マスが必ず入ります

以上のような感じですが，詳しい遊び方は8月号を参照しましょう。

なお，正解のチェックは，元の絵とは違う場合でもヒントとあっていれば正解となりますが，完璧な正解（元の絵とまったく同一の場合）とは，区別した表示を行います。でも，正解には違いないので，心配しないでね。ちゃんと正解として処理されます。

画面説明

このプログラムは，「EDITモード」と「PLAYモード」が，ワンキーで切り替えられるようになっています。すなわち，2つでひとつのプログラムとなっています。区別は，画面左上の表示でできます。なお，「PLAY」モード時には「WORLD」名は表示されません。当然ですね。また，自分のカーソル位置は，"O"で表されます。

操作方法

基本的に，2,4,6,8で，下，左，右，上にカーソルをそれぞれ動かしますが，それ以外にコマンドキーがあります。

●PLAYモード時

・Z……ドットを現在のカーソル位置に置く
・X……ドットを現在のカーソル位置から除去する
・E……EDITモードへ移る
・W……WORLD（面数）の変更
・C……正解かどうかの判定をする
・A……チェックマークをつける（正解の判定には関係ありません。プレイヤーが考えるときの手助けとしてお使いください）
・Q……ゲーム終了

●EDITモード時

・Z……ドットを現在のカーソル位置に置く
・X……ドットを現在のカーソル位置から除去する
・E……PLAYモードへ移る
・W……WORLD(面数)の変更
・F……エディットしているデータをクリア

・G……エディットしているデータを反転
・U……エディットしているエリアを1ドット分，上へ移動
・D……エディットしているエリアを1ドット分，下へ移動
・L……エディットしているエリアを1ドット分，左へ移動
・R……エディットしているエリアを1ドット分，右へ移動
・S……WORLD名入力(8文字まで認識)
・Q……ゲーム終了
・スペース……縦，横のドットの計算

WORLD変更画面の説明

表示される枠の中にある，“*”はまだクリアしたことのないWORLDで，“.”はすでにクリアしたステージです。

コマンドとしては，
・2,4,6,8……カーソル移動
・P……クリアしたWORLDをすべてクリアしてないことにする
・スペース……そのステージを選択
が使えます。

データファイルについて

オンメモリで格納しているので，WORLD DATAをエディットしたら，ゲームを終了させ，プロンプトに戻ってから，

#S PICT.DAT:E000:F15C:E000

で，セーブしてください。ゲームを始めるときは逆にデータファイルをメモリに読み込みましょう。オンメモリ方式を採っているため，データをモニタでのぞきつつ作ることもできます。

また，このデータファイルをたくさん作り，データをロードするごとにいくらでも面データを増やすこともできます（当たり前ですけどね)。

なお，データファイルのフォーマットは以下のとおりです。
・E000~E02B……ヘッダ
・E02C……WORLD DATA

表 変数表

X_dai：エディットエリアのデフォルトのX座標
Y_dai：エディットエリアのデフォルトのY座標
x：カーソルのX座標増分
y：カーソルのY座標増分
record：WORLDのクリアにかかった最短の時間
time：プレイしているワールドの時間
edit：1でエディットモード，0でプレイモード
X_old：カーソルを動かす前のX座標
Y_old：カーソルを動かす前のY座標
world：WORLDの値
WAITTIME：ウェイトタイムの値
BASE：データエリアの初期アドレス
ADR：ワールド名入力のテンポラリアドレス
play[256]：プレイエリアのデータ
fig[256]：エディットエリアのデータ
sub[16]：エディットエリアの移動の際のテンポラリ
pyo[128]：プレイ時の正解チェックのための横のデータ読み取り用
pta[128]：プレイ時の正解チェックのための縦のデータ読み取り用
yoko[128]：エディット時の正解チェックのための横のデータ読み取り用
tate[128]：エディット時の正解チェックのための縦のデータ読み取り用
cur_dat[]：カーソルのキャラクタデータ
msg_dat[][]：メッセージデータ

WORLD DATAは，44バイトで1WORLDとなっています。内訳は，
・1~32バイト……WORLDデータで，横一列が2バイトです。見たまんまの，左の1番目から8番目までを1バイトに圧縮しています。それが縦に16個分もあるので，2×16=32バイトです。
・33バイト……そのWORLDをクリアしているかのフラグです。1でクリア，0でまだクリアしていません
・34~35バイト……そのWORLDのRECORDデータです。1から65536までの値を取るので2バイトです
・36~44バイト……WORLD名を格納しています。8バイトが名前で，最後の1バイトが表示のストップデータです

なお，自分で新たなデータファイルを作りたいけど，わざわざクリアするのは面倒な方のために専用のデータクリアプログラムも一応用意しました。手抜きしているので，一応ある，という程度です。また，面データは，50WORLDまで自作のものを出させていただきます。

プログラムについて

あまり深く考えて作っていません。X68000版のソースをまったく参照してないので，無駄な部分が多くあるかもしれません。

そして，面データの格納アドレスを示しているのが，530行にあるBASEで，マシンによってスピード調節を行うのが，531行にあるWAITTIMEです。各自のマシンに合わせて調節してください。

S-OSではできなかったところ

うーがー！ と，最後までどうしようかと悩んだのは，X68000にあったクリアしたときに縮小表示される，というものです。MAGIC使わないとできそうにないし，ドット表示ってよくわかんないし，あっても解像度低いから苦しいし，ということでパスしてしまいました。ごめんなさい。次の機会までになんとかしましょう（MAGICの使い方，誰か教えて……しくしーく)。

リスト1

```
 1: //**********************************************************/
 2: //*                                                        */
 3: //*  P I C T   P U Z Z L E    O H ! X 投稿用  VER 1.000   */
 4: //*                                                        */
 5: //*    要S - O S   S L A N G                               */
 6: //*                                                        */
 7: //*                                   Presented By BCMA    */
 8: //**********************************************************/
 9: VAR X_dai,Y_dai,x,y,record,time,edit,X_old,Y_old,world;
10: VAR BASE,ADR,WAITTIME;
11: ARRAY BYTE play[256],fig[256],sub[16],yoko[128],tate[128];
12: ARRAY BYTE pyo[128],pta[128];
13: ARRAY BYTE cur_dat[2]=["O","@","0" ];
14: ARRAY BYTE msg_dat[18][13]=["            ¥0",
15:          "PICT Ver1.00¥0"," Puzzle     ¥0","     By BCMA¥0",
16:          " Stage_Name  ¥0","           ¥0","           ¥0",
17:          " Now Stage   ¥0","  Address   ¥0","/- Command --¥0",
18:          "/<U><D><L><R>¥0","/ ...move key¥0","/A...Marking ¥0",
19:          "/S...Stage in¥0","/W..World Sel¥0","/C..Calculate¥0",
20:          "/F..Fill     ¥0","/G..Reverse  ¥0","/E..Edit&Play¥0"];
21: MAIN()
22: BEGIN
23: PREP();
24: USER();
25: END;
26: //**********************************************************/
27: FIG1()
28: VAR I;
29: BEGIN
30: for I=0  to 23[
```

```
locate(13,I);print("I",SPC$(8),"I",SPC$(16),"I");
]
locate(14,8);print(STR$($2d,25));
 locate(14,0);print("--------");
 locate(15,1);print("WORLD");
 locate(15,4);print("RECORD");
 locate(16,6);print("TIME");
 locate(16,7);print(time);
for I=1  to  3[locate(0,I);print(msx$(msg_dat[I])); ]
for I=8  to 12[locate(0,I);print(msx$(msg_dat[I-4])); ]
for I=14 to 20[locate(0,I);print(msx$(msg_dat[I-5])); ]
END;
//***********************************************************/
FIG2()
VAR I;
BEGIN
locate(2,5);
if edit==1 [ print("Edit");locate(2,9);print(MSX$(ADR));
     ] else [ print("Play");locate(2,9);print("????????"); ]
locate(5,6);print("MODE");locate(16,7);print(SPC$(6));
time=0;
END;
//***********************************************************/
FIG3()
VAR I;
BEGIN
 locate(16,2);print(world,"  ");
 locate(16,5);print(spc$(6));locate(16,5);print(record);
locate(1,13);print(HEX4$(BASE+44)," - ",HEX4$(BASE+77));
for I=20 to 23[locate(0,I);print(msx$(msg_dat[I-5])); ]
END;
//***********************************************************/
USER()
VAR I,J,K,TEMP,ADREES;
BEGIN
x=8;y=7;ADREES=BASE+world*44;
CUR_PRINT();
locate(22,9);print("I");
LOOP:
if edit==1 [ EDITMODE(); ] else [
                   if edit==0 [ PLAYMODE(); ] else [ GOTO EX; ]
                                 ]
GOTO LOOP;
EX:
//データ格納
ADREES=BASE+world*44;
for I=1 to 16[
              MEMW[ADREES]=0;
     for J=0 to 15[
                    TEMP=0;
                    if fig[I*16+J-15]==1 [ TEMP=SET(TEMP,J); ]
                    MEMW[ADREES]=MEMW[ADREES]+TEMP;
                   ]
              ADREES=ADREES+2;
]
MEMW[ BASE+world*44+33 ]=record;locate(0,23);print(SPC$(13));
WIDTH(40);
END;
//***********************************************************/
PLAYMODE()
VAR WWW=0;
BEGIN
INKEY:
X_old=x;Y_old=y;WWW++;
if WWW==WAITTIME/2 [ locate(16,7);print(time++);WWW=0; ]
WAIT(WAITTIME);
CASE INKEY(0)[
           '4':[ if x<>1  [ x=x-1; ] ]
           '8':[ if y<>0  [ y=y-1; ] ]
           '2':[ if y<>15 [ y=y+1; ] ]
           '6':[ if x<>16 [ x=x+1; ] ]
       'q','Q':[ edit=2; ]
       'w','W':[ MOVE_WORLD(); ]
       'e','E':[ edit=1;CALC();CLEAR();FIG2(); ]
       'z','Z':[ play[x+y*16]=1; ]
       'x','X':[ play[x+y*16]=0; ]
       'c','C':[ SEIKAI(); ]
       'a','A':[ play[x+y*16]=2; ]
        OTHERS:[ GOTO INKEY; ]
             ]
CUR_PRINT();
WAIT(64);
if edit==0 [ GOTO INKEY; ]
END;
//***********************************************************/
EDITMODE()
VAR I;
BEGIN
INKEY:
X_old=x;Y_old=y;
WAIT(WAITTIME);
CASE INKEY(0)[
           '4':[ if x<>1  [ x=x-1; ] ]
           '8':[ if y<>0  [ y=y-1; ] ]
           '2':[ if y<>15 [ y=y+1; ] ]
           '6':[ if x<>16 [ x=x+1; ] ]
       'q','Q':[ edit=2; ]
       'w','W':[ MOVE_WORLD(); ]
       'z','Z':[ fig[x+y*16]=1; ]
       'x','X':[ fig[x+y*16]=0; ]
       's','S':[ STAGE(); ]
           ' ':[ CALC(); ]
       'f','F':[ FILL_GRA(); ]
       'g','G':[ REVERSE(); ]
       'U','u':[ FIG_UP(); ]
       'D','d':[ FIG_DOWN(); ]
       'R','r':[ FIG_RIGHT(); ]
       'L','l':[ FIG_LEFT(); ]
       'e','E':[ edit=0;CLEAR();CALC();FIG2();
                 for I=0 to 256[ play[I]=0; ]
             ]
        OTHERS:[ GOTO INKEY; ]
             ]
CUR_PRINT();
WAIT(64);
END;
//***********************************************************/
CUR_PRINT()
VAR I;
BEGIN
locate(X_dai+X_old-1,8);print("-");
locate(22,Y_dai+Y_old);print("I");
LOCATE(27,8);PRINT("+");
LOCATE(32,8);PRINT("+");
LOCATE(37,8);PRINT("+");
LOCATE(22,13);PRINT("+");
LOCATE(22,18);PRINT("+");
LOCATE(22,23);PRINT("+");
locate(13,24);print("I");
locate(22,24);print("I");
locate(X_old+X_dai-1,Y_old+Y_dai);
I=Y_old*16+X_old;
if edit==1 [
if fig[I]==1 [ print(STR$($7B,1)); ] else [ print(" ");]
locate(x+X_dai-1,y+Y_dai);print(STR$(cur_dat[fig[y*16+x]],1));
           ] else [
if play[I]==1 [ print(STR$($7B,1)); ] else [
if play[I]==2 [ print(STR$(45,1)); ] else [ print(" "); ]
locate(x+X_dai-1,y+Y_dai);
print(STR$(cur_dat[play[y*16+x]],1));
                                             ]
locate(x+X_dai-1,y+Y_dai);
print(STR$(cur_dat[play[y*16+x]],1));
           ]
locate(x+X_dai-1,8);print("@");locate(22,y+Y_dai);print("@");
END;
//***********************************************************/
CALC()
VAR I,J,M,L;
VAR CH;
BEGIN
for I=0 to 15[ locate(14,9+I);print(SPC$(8)); ]
for I=0 to 7[ locate(23,0+I);print(SPC$(16)); ]
//****************横******************/
for J=1 to 16 [ CH=0;M=0;L=21;locate(L,24-J+1);
 for I=1 to 16 [
  if fig[ 273-J*16-I ]==1 [ M++; ] else [
   if M<>0 [ if M<10 [ print(M); ] else [ print(CHR$(55+M)); ]
            M=0;L--;locate(L,25-J);CH=1;
            ]
                                         ]
               ]
if M<>0 [ CH=1;if M<10 [ print(M); ] else [ print(CHR$(55+M)); ] ]
IF CH==0 [ LOCATE(L,25-J);PRINT("0"); ]
]
//****************縦******************/
for J=1 to 16 [ CH=0;M=0;L=7;locate(39-J,L);
 for I=1 to 16 [
  if fig[ 273-I*16-J ]==1 [ M++; ] else [
   if M<>0 [ if M<10 [ print(M);] else [ print(CHR$(55+M)); ]
            M=0;L--;locate(39-J,L);CH=1;
            ]
                                         ]
               ]
if M<>0 [ CH=1;if M<10 [ print(M);] else [ print(CHR$(55+M)); ] ]
IF CH==0 [ LOCATE(39-J,L);PRINT("0"); ]
]
END;
//***********************************************************/
STAGE()
VAR I,J,ADREES;
BEGIN
ADREES=BASE+world*44+35;
locate(0,20);print("-INPUT STAGE-");
locate(0,21);print("     NAME    ");
locate(0,22);print(SPC$(13));
locate(0,23);print(SPC$(13));
LOCATE(0,22);I=GETLIN(ADR,8);
//*９８版ではここのlocate(0,22)をlocate(0,0)に変更してね！*//
//*また、ステージ名入力がうまくいかない人も変更してみそ！*//
WIDTH(40);FIG1();FIG2();
if I<>-1 [
            locate(2,9);PRINT(SPC$(8));
            locate(2,9);PRINT(MSX$(ADR));
            for J=0 to 7[ MEM[ADREES+J]=MEM[ADR+J]; ]
         ]
FIG3();
END;
//***********************************************************/
MOVE_WORLD()
VAR I,J,K,TEMP,ADREES;
BEGIN
ADREES=BASE+world*44;     //32+2+8+1+1
//データ格納
for I=1 to 16[
              MEMW[ADREES]=0;
     for J=0 to 15[
                    TEMP=0;
                    if fig[I*16+J-15]==1 [ TEMP=SET(TEMP,J); ]
                    MEMW[ADREES]=MEMW[ADREES]+TEMP;
                   ]
              ADREES=ADREES+2;
]
MEMW[ BASE+world*44+33 ]=record;locate(0,23);print(SPC$(13));
STAGE_FIG();
FIG3();
for I=0 to 256[ play[I]=0;fig[I]=0; ]
for J=0 to 8[ MEM[ADR+J]=0; ]
//データ取りだし
ADREES=BASE+world*44;TEMP=ADREES;
for I=1 to 16[
 for J=0 to 15[ fig[I*16+J-15]=BIT(MEMW[ADREES],J); ]
 ADREES=ADREES+2;
]
if MEM[ BASE+world*44+32 ]==0 [
                  record=9999; ] else [ record=MEMW[++ADREES]; ]
locate(16,5);print(record);time=0;ADREES=BASE+world*44+35;
CLEAR();
locate(1,13);print(HEX4$(TEMP)," - ",HEX4$(TEMP+43));
for J=0 to 7[ MEM[ADR+J]=MEM[ADREES+J]; ]
locate(2,9);PRINT(SPC$(8));locate(2,9);
if edit==1 [ PRINT(MSX$(ADR)); ] else [ PRINT("????????"); ]
locate(16,7);print(SPC$(6));
CALC();
END;
//***********************************************************/
```

▶うーむ。そろそろCD-ROMドライブがほしくなってきました。速くて安いドライブってなんでしょうか？　Zipドライブも捨てがたいし……。　加藤　雅浩(26)岡山県

```
STAGE_FIG()
VAR I,J,X_sel=22,Y_sel=14,xx=2,yy=2,OK=1,temp,ADREES;
ARRAY BYTE SELECT[100];
BEGIN
X_old=xx;Y_old=yy;
locate(13,24);print(SPC$(26));
for I=0 TO 23[ locate(14,I);print(SPC$(25)); ]
for I=0 to 9[
            locate(X_sel-1,Y_sel+I);print("I",SPC$(10),"I");
            locate(X_sel+I,Y_sel-1);print(I+1);
            locate(X_sel-2,Y_sel+I);print(I);
]
for J=0 to 7[ MEM[ADR+J]=MEM[ BASE+world*44+35+J ]; ]
MEM[ADR+8]=0;
locate(32,4);print(SPC$(6));
locate(16,1);print("--< World Selecter >--");
locate(17,3);print(" World Number   ",world,"  ");
locate(17,4);print(" World Record   ",MEMW[BASE+world*44+33]);
if edit==1 [ locate(17,5);print(" World Name  ",MSX$(ADR));]
locate(17,7);print("   8              ");
locate(17,8);print(" 4   6  Cursole Move");
locate(17,9);print("   2   ");
locate(17,10);print("  SPC.....select");
locate(17,11);print("   P.....flag CLEAR");
ADREES=BASE+44+32;
for I=0 to 9[
  for J=1 to 10[
      SELECT[I*9+J]=MEM[ADREES];
      ADREES=ADREES+44;
      locate(X_sel+J-1,Y_sel+I);
      if SELECT[I*10+J]==1 [print(".");] else [print("*");]
]
]
locate(xx+X_sel-1,yy+Y_sel);
if SELECT[ world ]==1 [ print("O"); ] else [ print("@"); ]
III:
WAIT(WAITTIME);
CASE INKEY(0)[
         '4':[ if xx<>1  [ xx=xx-1; ] ]
         '8':[ if yy<>0  [ yy=yy-1; ] ]
         '2':[ if yy<>9  [ yy=yy+1; ] ]
         '6':[ if xx<>10 [ xx=xx+1; ] ]
         'P':[ for I=1 to 100[ SELECT[I]=0; ]
               ADREES=BASE+44+32;
               for I=0 to 9[for J=1 to 10[
               MEM[ADREES]=SELECT[I*9+J];ADREES=ADREES+44;
               locate(X_sel+J-1,Y_sel+I);print("*"); ]]
             ]
         ' ':[ GOTO EX2; ]
     OTHERS:[ GOTO III; ]
             ]
WAIT(64);
world=yy*10+xx;
for J=0 to 7[ MEM[ADR+J]=MEM[ BASE+world*44+35+J ]; ]
locate(32,4);print(SPC$(6));
locate(17,3);print(" World Number   ",world,"  ");
locate(17,4);print(" World Record   ",MEMW[BASE+world*44+33]);
if edit==1 [ locate(17,5);print(" World Name  ",MSX$(ADR));]
locate(X_old+X_sel-1,Y_old+Y_sel);
if SELECT[Y_old*10+X_old]==1 [print(".");] else [print("*");]
locate(xx+X_sel-1,yy+Y_sel);
if SELECT[ world ]==1 [ print("O"); ] else [ print("@"); ]
X_old=xx;Y_old=yy;
GOTO III;
EX2:
FIG1();
END;
//************************************************************/
FIG_UP()
VAR I,J;
BEGIN
for I=1 to 16[
            sub[I]=fig[I];
            for J=1 to 15[ fig[I+J*16-16]=fig[I+J*16]; ]
            fig[240+I]=sub[I];
]
CLEAR();
END;
//************************************************************/
FIG_DOWN()
VAR I,J;
BEGIN
for I=1 to 16[
           sub[I]=fig[240+I];
           for J=1 to 15[ fig[256+I-J*16]=fig[240+I-J*16];]
           fig[I]=sub[I];
]
CLEAR();
END;
//************************************************************/
FIG_RIGHT()
VAR I,J;
BEGIN
for I=1 to 16[
            sub[I]=fig[I*16];
            for J=1 to 15[ fig[I*16-J+1]=fig[I*16-J]; ]
            fig[I*16-15]=sub[I];
]
CLEAR();
END;
//************************************************************/
FIG_LEFT()
VAR I,J;
BEGIN
for I=1 to 16[
            sub[I]=fig[I*16-15];
            for J=1 to 15[ fig[I*16-16+J]=fig[I*16+J-15]; ]
            fig[I*16]=sub[I];
]
CLEAR();
END;
//************************************************************/
FILL_GRA()
VAR I;
BEGIN
for I=1 to 256[ fig[I]=0; ]
CLEAR();CALC();
END;
//************************************************************/
REVERSE()
VAR I;
BEGIN
for I=1 to 256[ if fig[I]==0 [ fig[I]=1; ] else [ fig[I]=0;] ]
CLEAR();CALC();
END;
//************************************************************/
SEIKAI()
VAR I,J,LOST,M,II;
BEGIN
II=1;
for I=1 to 128[ tate[I]=0;yoko[I]=0;pta[I]=0;pyo[I]=0; ]
//*****************横play*****************/
for J=0 to 15 [ M=0;
 for I=1 to 16 [
  if play[J*16+I]==1 [M++;] else [if M<>0 [pyo[II++]=M;M=0;] ]
               ]
if M<>0 [ pyo[II++]=M; ]
]
II=1;
//*****************縦play*****************/
for J=1 to 16 [ M=0;
 for I=0 to 15 [
  if play[I*16+J]==1 [M++;] else [if M<>0 [pta[II++]=M;M=0;] ]
               ]
if M<>0 [ pta[II++]=M; ]
]
II=1;
//*****************横fig*****************/
for J=0 to 15 [ M=0;
 for I=1 to 16 [
  if fig[J*16+I]==1 [M++;] else [if M<>0 [yoko[II++]=M;M=0;] ]
               ]
if M<>0 [ yoko[II++]=M; ]
]
II=1;
//*****************縦fig*****************/
for J=1 to 16 [ M=0;
 for I=0 to 15 [
  if fig[I*16+J]==1 [M++;] else [if M<>0 [tate[II++]=M;M=0;] ]
               ]
if M<>0 [ tate[II++]=M; ]
]
II=0;LOCATE(3,10);
for I=1 to 128[
             if tate[I]<>pta[I] [ II=1; ]
             if yoko[I]<>pyo[I] [ II=1; ]
]
if II==1 [ LOST(); ] else [ PERFECT(); ]
END;
//************************************************************/
PERFECT()
VAR I,J,K,L,xx,yy,LOST;
BEGIN
WIDTH(40);
xx=X_dai+1;yy=Y_dai+1;LOST=0;
for K=0 to 5[
yy--;
    for L=0 to 3[
            for I=0 to 16[ locate(xx,yy+I);print(SPC$(16));]
xx--;
for I=0 to 15[ for J=1 to 16[
  if play[I*16+J]==1 [locate(xx+J-1,yy+I);print(STR$($7B,1));]
                       ]]
]]
BEEP();BEEP();
for I=0 to 15 [
 for J=1 to 16 [
  if fig[I*16+J]==1 [ if play[I*16+J]<>1 [ LOST=1;I=15;J=16; ]
           ] else [ if play[I*16+J]==1 [ LOST=1;I=15;J=16; ]
                  ]
]]
locate(23,14);
if LOST==1 [ print(" Normal CLEAR");
    ] else [ print("   PERFECT!  "); ]
locate(22,12);
PRINT(world,"  World ",MSX$(BASE+world*44+35));
MEM[ BASE+world*44+32 ]=1;
if record>time [ record=time;
              locate(20,17);print("You get new record!");
              locate(28,19);print(record);
               ]
locate(9,1);print("PICT PUZZLE for S-OS");
locate(18,5);print("Thank you");
locate(25,6);print("your playing!");
locate(24,9);print("GOOD JOB !!");
HIT_ANY(16);WIDTH(40);FIG1();FIG2();FIG3();CALC();
for I=1 to 256 [ play[I]=0; ]
END;
//************************************************************/
LOST()
BEGIN
BEEP();
locate(0,21);print(spc$(13));locate(0,23);print(spc$(13));
locate(0,22);print(spc$(13));locate(5,22);print("LOST");
HIT_ANY(2);
END;
//************************************************************/
WAIT(X)
VAR I,J;
BEGIN
for I=1 to 4 [ for J=1 to X [ ]  ]
END;
//************************************************************/
CLEAR()
VAR I,J;
BEGIN
for I=1 to 16[ locate(23,8+I);print(SPC$(16)); ] //BOX CLEAR
if edit==1 [
   for I=0 to 15[
       for J=1 to 16[
           if fig[I*16+J]==1 [
               locate(X_dai+J-1,Y_dai+I);print(STR$($7B,1));
                             ]
                  ]
              ]
]
```

▶道を歩いていると猫がいたのですが，その猫は私に気づくと素早くどぶ板の下に潜り込んでしまいました。別になんにもしないのに。なんでそんなところに逃げるかなー。
竹本　郁馬(24)神奈川県

```
END;
//***************************************************************/
HIT_ANY(Z)
VAR KEY,I;
BEGIN
locate(0,21);print(SPC$(13));locate(0,23);print(SPC$(13));
        REPEAT[
                I=RND(6)+1;
                KBUF_CLR();
                LOCATE(Z,23);PRINT("HIT ANY !!");
                LOCATE(Z,23);PRINT(SPC$(10));
                IF (KEY=INKEY(0)) == $1B
                STOP();
        ] UNTIL KEY != $00;
FIG3();
END;
//***************************************************************/
KBUF_CLR()
BEGIN
     WHILE INKEY(0) != 0 [
     ]
END;
```

```
//***************************************************************/
PREP()
VAR ADREES,I,J;
BEGIN
WIDTH(40);
record=9999;time=0,edit=1;world=1;X_dai=23;Y_dai=9;BASE=$E000;
ADR=BASE+30;WAITTIME=210;
//*ここのWAITTIME=10の値は各自のマシンにあわせて調節してね*/
for I=0 to 256 [ play[I]=0;fig[I]=0; ]
FIG1();FIG2();FIG3();
//データ取りだし
ADREES=BASE+world*44;
for I=1 to 16[
 for J=0 to 15[ fig[I*16+J-15]=BIT(MEMW[ADREES],J); ]
 ADREES=ADREES+2;
]
END;
//***************************************************************/
//*                                                             */
//*    さあ、ＰＩＣＴパズルコンテストにみんなで、応募だぁ！     */
//*                                                             */
//***************************************************************/
```

リスト2

```
E000 50 49 43 54 50 54 5A 5A : 88
E008 4C C5 00 44 41 54 41 00 : 2B
E010 42 59 42 43 4D 41 00 00 : AE
E018 00 00 00 00 00 00 53 54 : A7
E020 41 52 20 20 20 20 00 00 : 13
E028 00 00 00 00 00 00 00 00 : 00
E030 1F 00 00 00 00 00 00 00 : 1F
E038 00 00 00 00 00 00 00 00 : 00
E040 00 00 00 00 00 00 00 00 : 00
E048 00 00 00 00 00 0F 27 4C : 82
E050 49 4E 45 20 20 20 20 00 : 5C
E058 04 00 04 00 1F 00 04 00 : 2B
E060 04 00 00 00 00 00 00 00 : 04
E068 00 00 00 00 00 00 00 00 : 00
E070 00 00 00 00 00 00 00 00 : 00
E078 00 0F 27 43 52 4F 53 53 : C0
--------------------------------------
SUM: 8F 16 15 5E 8F 87 8C 4D 5760

E080 20 20 20 00 04 00 0E 00 : 72
E088 1F 00 0E 00 04 00 00 00 : 31
E090 00 00 00 00 00 00 00 00 : 00
E098 00 00 00 00 00 00 00 00 : 00
E0A0 00 00 00 00 00 0F 27 48 : 7E
E0A8 4F 53 48 49 20 20 20 00 : 93
E0B0 00 00 0E 00 0E 00 0E 00 : 2A
E0B8 00 00 00 00 00 00 00 00 : 00
E0C0 00 00 00 00 00 00 00 00 : 00
E0C8 00 00 00 00 00 00 00 00 : 00
E0D0 00 0F 27 42 4F 58 20 20 : 5F
E0D8 20 20 20 00 1F 00 11 00 : 90
E0E0 11 00 11 00 1F 00 00 00 : 41
E0E8 00 00 00 00 00 00 00 00 : 00
E0F0 00 00 00 00 00 00 00 00 : 00
E0F8 00 00 00 00 00 0F 27 57 : 8D
--------------------------------------
SUM: BF A2 DC 8B C3 96 BB BF D828

E100 41 4B 55 20 20 20 20 00 : 61
E108 1B 00 0A 00 04 00 0A 00 : 33
E110 1B 00 00 00 00 00 00 00 : 1B
E118 00 00 00 00 00 00 00 00 : 00
E120 00 00 00 00 00 00 00 00 : 00
E128 00 0F 27 50 45 4B 45 20 : 7B
E130 20 20 20 00 1B 00 11 00 : 8C
E138 00 00 11 00 1B 00 00 00 : 2C
E140 00 00 00 00 00 00 00 00 : 00
E148 00 00 00 00 00 00 00 00 : 00
E150 00 00 00 00 00 0F 27 41 : 77
E158 53 54 45 52 4F 49 44 00 : 1A
E160 15 00 0A 00 15 00 0A 00 : 3E
E168 15 00 00 00 00 00 00 00 : 15
E170 00 00 00 00 00 00 00 00 : 00
E178 00 00 00 00 00 00 00 00 : 00
--------------------------------------
SUM: 14 CE 06 C2 03 C3 F5 61 EE0E

E180 00 0F 27 43 48 45 43 4B : 94
E188 45 52 20 00 0E 00 15 00 : DA
E190 0E 00 0A 00 11 00 00 00 : 29
E198 00 00 00 00 00 00 00 00 : 00
E1A0 00 00 00 00 00 00 00 00 : 00
E1A8 00 00 00 00 00 0F 27 49 : 7F
E1B0 4E 56 41 44 45 52 20 00 : E0
E1B8 0E 00 1F 00 1F 00 17 00 : 63
E1C0 0E 00 00 00 00 00 00 00 : 0E
E1C8 00 00 00 00 00 00 00 00 : 00
E1D0 00 00 00 00 00 00 00 00 : 00
E1D8 00 0F 27 42 41 4C 4C 20 : 71
E1E0 20 20 20 00 40 00 E0 00 : 80
E1E8 F0 01 FF 1F FE 0F FC 07 : 1F
E1F0 F8 03 BC 07 0E 0E 07 1C : FD
E1F8 00 00 00 00 00 00 00 00 : 00
--------------------------------------
SUM: C5 EA B3 EF 58 0F E5 D7 C8EC

E200 00 00 00 00 00 0F 27 53 : 89
E208 54 41 52 20 20 20 20 00 : 67
E210 E0 00 FE 0F 08 02 08 02 : 01
E218 FF 1F 00 00 FE 0F 02 08 : 35
E220 FE 0F 02 08 FE 0F 00 00 : 24
E228 00 00 00 00 00 00 00 00 : 00
E230 00 0F 27 4F 54 4F 20 20 : 68
E238 20 20 20 00 00 00 F8 07 : 5F
E240 FC 0F 1C 0E 1C 0E FC 0F : 6A
E248 FC 07 0C 00 0C 00 0C 00 : 27
E250 1E 00 00 00 00 00 00 00 : 1E
E258 00 00 00 00 00 0F 27 46 : 7C
E260 4C 41 47 20 20 20 20 00 : 54
E268 00 00 44 04 EE 0E 44 04 : 8C
E270 00 00 FC 03 FE 07 FC 03 : 03
E278 F8 01 00 00 00 00 00 00 : F9
--------------------------------------
SUM: AB F6 48 BB AC F0 F8 E0 EDB1

E280 00 00 00 00 00 00 00 00 : 00
E288 00 0F 27 41 53 48 49 41 : 9C
E290 54 4F 20 00 40 00 E0 00 : E3
E298 F0 01 F8 03 FF 1F 40 00 : 4A
E2A0 40 00 40 00 40 01 C0 01 : 82
E2A8 00 00 00 00 00 00 00 00 : 00
E2B0 00 00 00 00 00 0F 27 4B : 81
E2B8 41 53 41 20 20 20 20 00 : 55
E2C0 00 00 98 19 30 33 98 19 : C5
E2C8 30 33 9B D9 31 B3 03 C0 : 7E
E2D0 06 70 FC 3F 00 00 00 00 : B1
E2D8 00 00 00 00 00 00 00 00 : 00
E2E0 00 0F 27 4F 4E 53 45 4E : B9
E2E8 20 20 20 00 80 00 80 00 : 60
E2F0 C0 01 F0 03 98 06 98 0C : F6
E2F8 F0 08 60 0C 00 06 00 03 : 6D
--------------------------------------
SUM: CB 8D 86 F3 B9 DC 68 C3 359E

E300 80 01 F0 0F FC 3F FE 7F : 38
E308 FE 7F FE 7F 00 0F 27 53 : 83
E310 48 49 54 54 50 4F 20 00 : F8
E318 00 00 80 03 80 07 80 0C : 96
E320 80 18 80 13 80 07 80 0C : 3E
E328 80 18 80 10 F0 00 F8 00 : 10
E330 F8 00 70 00 00 00 00 00 : 68
E338 00 0F 27 4F 4E 50 55 20 : 98
E340 20 20 20 00 00 00 F8 1F : 77
E348 0C 30 04 20 0C 30 38 1F : F3
E350 00 01 00 01 00 01 00 01 : 04
E358 00 00 C0 03 E0 07 C0 03 : 6D
E360 00 00 00 00 00 0F 27 48 : 7E
E368 41 54 45 4E 41 20 20 00 : A9
E370 80 00 80 03 80 06 80 03 : 0C
E378 F8 00 8C 00 E6 07 96 0C : 13
--------------------------------------
SUM: A3 AD 8E CC 1D 6F DF A3 5A09

E380 E6 09 8C 0C 98 06 F0 03 : 18
E388 80 00 80 18 80 0C 80 07 : 2B
E390 00 0F 27 54 4F 20 4F 4E : 96
E398 20 20 20 00 80 00 FE 3F : 1D
E3A0 80 00 FE 3F 82 20 FE 3F : 9C
E3A8 82 20 FE 3F C0 01 E0 03 : 83
E3B0 B0 06 98 0C 8C 18 86 70 : F4
E3B8 00 00 00 00 00 0F 27 48 : 7E
E3C0 49 47 41 53 48 49 20 00 : D5
E3C8 00 00 80 01 FE 7F 80 01 : 7F
E3D0 80 01 FE 7F 02 40 62 46 : E8
E3D8 C2 43 E2 47 82 41 E2 47 : 1A
E3E0 82 41 82 41 00 00 00 00 : 86
E3E8 00 0F 27 4D 49 4E 41 4D : A8
E3F0 49 20 20 00 00 00 00 00 : 89
E3F8 00 00 FE 7F 40 02 40 02 : 01
--------------------------------------
SUM: 8E 59 4F 29 08 13 AD 6E 154D

E400 40 02 FE 7F 42 42 62 46 : EB
E408 3A 5C 02 40 02 40 02 40 : 5C
E410 FE 7F 00 00 00 0F 27 4E : 01
E418 49 53 48 49 20 20 20 00 : 8D
E420 00 00 20 02 20 02 20 02 : 66
E428 3E 7E 20 02 20 02 20 02 : 22
E430 20 02 20 02 20 02 70 46 : 1C
E438 18 44 0E 7C 00 00 00 00 : E6
E440 00 0F 27 50 45 49 20 20 : 54
E448 20 20 20 00 00 00 00 00 : 60
E450 00 00 FE 3F 82 20 82 20 : 81
E458 82 20 82 20 FE 3F 82 20 : 23
E460 82 20 82 20 82 20 FE 3F : 23
E468 00 00 00 00 00 0F 27 54 : 8A
E470 41 4E 4E 42 4F 20 20 00 : AE
E478 00 00 22 3F 36 30 3E 18 : 1D
--------------------------------------
SUM: 9C B1 6F DA 90 DE 02 29 333E

E480 2A 0C 22 06 22 3F 00 00 : BF
E488 00 00 DE 4B 52 6A 52 3A : 71
E490 4E 2A 52 6A 52 4A DE 4B : F9
E498 00 0F 27 4D 5A 20 38 30 : 65
E4A0 4B 20 20 00 00 00 7E 7E : 87
E4A8 40 40 40 40 7E 7C 40 06 : 40
E4B0 40 02 7E 7E 00 00 02 00 : 40
E4B8 82 08 02 3C 9E 08 92 28 : 28
E4C0 9E 38 00 00 00 0F 27 33 : 3F
E4C8 32 42 49 54 20 20 20 00 : 71
E4D0 00 00 F8 07 08 04 08 00 : 13
E4D8 F8 07 00 04 08 04 F8 07 : 0E
E4E0 00 00 E0 7B 20 4A 20 0A : EF
E4E8 2E 7A 20 42 20 4A E0 7B : CF
E4F0 00 0F 27 53 2D 4F 53 20 : 78
E4F8 20 20 20 00 00 00 80 00 : E0
--------------------------------------
SUM: DB D9 E1 71 D9 B1 D4 40 DD2B

E500 80 00 C0 01 C0 01 C0 01 : C3
E508 C0 01 C0 01 C0 01 C0 01 : 04
E510 C0 01 F8 0F 40 01 C0 01 : CA
E518 40 01 C0 01 00 0F 27 53 : 8B
E520 57 4F 52 44 20 20 20 00 : 9C
E528 00 00 00 00 7F FE FE 7C : F7
E530 FC 39 F8 13 F0 07 E0 0F : 26
E538 C8 1F 9C 3F 3E 7F 7F FE : FC
E540 FF FC 00 00 00 00 00 00 : FB
E548 00 0F 27 58 20 20 20 20 : 0E
E550 20 20 20 00 00 00 FE 7F : DD
E558 FE 7F 7E 3E 3E 7C 7E 7E : EF
E560 FE 7B FE 7F 7E 7E 7E 7E : EE
E568 FE 7F 00 00 00 00 00 00 : 7D
E570 00 00 00 00 00 0F 27 46 : 7C
E578 4C 4F 4F 50 59 20 20 00 : D3
--------------------------------------
SUM: C0 9D 30 0D C2 FF 45 C0 0797

E580 00 00 E0 03 20 00 20 00 : 23
E588 E0 01 20 00 E0 00 20 00 : 01
E590 20 00 20 00 70 00 DC 01 : 8D
E598 8C 01 DC 01 F8 00 00 00 : 62
E5A0 00 0F 27 4B 45 59 20 20 : 5F
E5A8 20 20 20 00 00 00 00 00 : 60
E5B0 06 11 84 19 8C 08 98 0C : EC
E5B8 00 00 FC 1F F8 0F F8 0F : 29
E5C0 98 0F 90 07 30 07 20 03 : 98
E5C8 E0 03 00 00 00 0F 27 43 : 5C
E5D0 52 41 4B 45 52 20 20 00 : B5
E5D8 E0 03 38 0E C8 09 B8 0E : C0
E5E0 80 00 80 30 9C 18 B0 0C : A0
E5E8 E0 0E E0 07 FE 3F FC 1F : 2D
E5F0 F8 0B F8 09 38 08 F8 0F : 4B
E5F8 00 0F 27 46 4C 4F 57 45 : B3
--------------------------------------
SUM: B4 C0 55 67 99 5D E6 0F F8F6

E600 52 20 20 00 00 00 00 00 : 92
E608 F0 07 18 0C 0C 18 04 10 : 53
E610 04 10 04 10 0C 18 18 0C : 70
E618 30 06 22 22 22 22 3E 3E : 3A
E620 00 00 00 00 00 0F 27 41 : 77
E628 55 4D 20 20 20 20 20 00 : 42
E630 00 00 00 00 FE 7F 02 40 : BF
E638 02 40 FA 4F 02 40 FA 4F : 16
E640 82 40 82 40 82 40 02 40 : 88
```

▶大家に間違いの電話代を８万円請求されたあと，自分の回線を引くことになり，加入料と工事費で結局８万円出費した。　和田　智(20)長野県

```
E648 02 40 FE 7F 00 00 00 00 : BF
E650 00 0F 27 59 55 42 49 4E : BD
E658 20 20 20 00 00 00 00 00 : 60
E660 78 0F 8C 18 08 08 0C 18 : 5F
E668 68 0B 2C 1A E8 0B 0C 18 : D0
E670 A8 0B 0C 18 08 08 0C 18 : 0B
E678 F8 0F 00 00 00 0F 27 43 : 80
---------------------------------
SUM: F1 AD 03 0F 29 EC 33 43 EB2B

E680 50 55 20 20 20 20 20 00 : 45
E688 00 00 FC 7F 04 40 84 43 : 86
E690 C4 47 84 43 04 40 FC 7F : 91
E698 04 00 04 00 04 00 04 00 : 10
E6A0 04 00 04 00 04 00 04 00 : 10
E6A8 00 0F 27 48 49 4E 4F 4D : B1
E6B0 41 52 55 00 E0 03 F0 07 : C2
E6B8 D8 0F D8 0F F8 0F F0 07 : CC
E6C0 00 00 E0 03 F0 07 C0 00 : 9A
E6C8 40 00 E0 01 30 03 E0 01 : 35
E6D0 C0 00 80 01 00 0F 27 50 : C7
E6D8 41 4E 43 48 49 20 20 00 : A3
E6E0 00 00 FE 7E 82 42 BA 5A : 54
E6E8 82 42 82 42 82 42 82 42 : 10
E6F0 AA 42 AA 42 AA 42 AA 42 : B0
E6F8 82 42 FE 7E 82 42 FE 7F : 81
---------------------------------
SUM: 24 20 A7 06 EA 41 A2 CB 2B2F

E700 00 0F 27 4D 41 4E 48 41 : 9B
E708 54 41 4E 00 00 00 FE 01 : E2
E710 F8 03 E0 07 C0 0F 80 1F : 50
E718 00 3F 00 76 00 76 00 3F : 6A
E720 80 1F C0 0F E0 07 F0 03 : 48
E728 FE 01 00 00 00 0F 27 42 : 77
E730 55 4D 45 52 41 4E 20 00 : E8
E738 00 00 78 00 F8 01 F8 07 : 70
E740 F8 0F F8 1F F8 1F F8 3F : 6C
E748 F8 3F F8 1F F8 1F F8 0F : 6C
E750 F8 07 F8 01 78 00 00 00 : 70
E758 00 0F 27 48 41 52 46 4D : A4
E760 4F 4F 4E 00 00 00 18 0C : 10
E768 30 06 60 03 E0 03 30 06 : B2
E770 18 0C 30 06 E0 03 80 00 : BD
E778 FC 1F 80 00 80 00 80 00 : 9B
---------------------------------
SUM: 9A E3 3F BB 03 CE 73 99 8B72

E780 00 00 00 00 00 0F 27 4D : 83
E788 45 52 43 55 52 59 20 00 : FA
E790 00 00 00 00 F0 07 18 0C : 1B
E798 08 08 18 0C F0 07 80 00 : AB
E7A0 80 00 FE 3F 80 00 80 00 : BD
E7A8 80 00 80 00 00 00 00 00 : 00
E7B0 00 0F 27 56 45 4E 55 53 : C7
E7B8 20 20 20 00 00 00 00 00 : 60
E7C0 80 00 C0 01 E0 03 C0 01 : E5
E7C8 80 00 00 00 04 20 1C 38 : F8
E7D0 78 1E F0 0F C0 03 00 00 : 58
E7D8 00 00 00 00 00 0F 27 4D : 83
E7E0 4F 4F 4E 20 20 20 20 00 : 6C
E7E8 00 00 00 78 00 70 00 58 : 40
E7F0 00 0C 00 06 00 03 80 01 : 96
E7F8 C0 00 7E 00 C3 00 81 00 : 82
---------------------------------
SUM: F4 02 9C A4 7E 8C D8 8B B993

E800 81 00 C3 00 7E 00 00 00 : C2
E808 00 0F 27 4D 41 52 53 20 : 89
E810 20 20 20 00 00 00 00 02 : 62
E818 00 02 0C 02 18 02 30 02 : 5C
E820 20 02 30 02 18 02 0C 02 : 7C
E828 06 02 FF 3F 00 02 00 02 : 4A
E830 00 02 00 00 00 0F 27 4A : 82
E838 55 50 49 54 45 52 20 00 : F9
E840 00 00 5E 77 52 21 5E 21 : C7
E848 42 27 00 00 00 38 20 29 : EA
E850 30 3B 18 06 00 00 00 00 : 89
E858 5E 15 08 54 1C 76 16 33 : AA
E860 00 0F 27 50 49 43 54 50 : B6
E868 55 5A 5A 00 00 00 00 20 : 29
E870 00 F0 00 20 00 00 00 30 : 40
E878 00 72 00 70 00 74 F0 7D : C3
---------------------------------
SUM: 41 C9 8D 95 EB 3F AE 0C 92D1

E880 F8 7F 06 7F 66 3E 66 28 : 2E
E888 00 00 00 00 00 0F 27 4A : 80
E890 41 50 41 4E 20 20 20 00 : 80
E898 00 00 F8 0F 0C 18 0C 18 : 4F
E8A0 F8 0F 40 00 43 40 46 E0 : F0
E8A8 FD 7B 58 0E 40 00 E0 00 : FE
E8B0 B0 01 19 87 0F CC 07 78 : AB
E8B8 00 0F 27 4B 55 48 41 4B : AA
E8C0 4A 49 4E 00 00 00 00 00 : E1
E8C8 00 00 00 00 00 00 00 00 : 00
E8D0 00 00 00 00 00 00 00 00 : 00
E8D8 00 00 00 00 00 00 00 00 : 00
E8E0 00 00 00 00 00 0F 27 20 : 56
E8E8 20 20 20 20 20 20 20 00 : E0
E8F0 00 00 00 00 00 00 00 00 : 00
E8F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 48 D2 85 DC 99 08 6E 4D 8BE2

E900 00 00 00 00 00 00 00 00 : 00
E908 00 00 00 00 00 00 00 00 : 00
E910 00 0F 27 20 20 20 20 20 : D6
E918 20 20 20 00 00 00 00 00 : 60
E920 00 00 00 00 00 00 00 00 : 00
E928 00 00 00 00 00 00 00 00 : 00
E930 00 00 00 00 00 00 00 00 : 00
E938 00 00 00 00 00 0F 27 20 : 56
E940 20 20 20 20 20 20 20 00 : E0
E948 00 00 00 00 00 00 00 00 : 00
E950 00 00 00 00 00 00 00 00 : 00
E958 00 00 00 00 00 00 00 00 : 00
E960 00 00 00 00 00 00 00 00 : 00
E968 00 0F 27 20 20 20 20 20 : D6
E970 20 20 20 00 00 00 00 00 : 60
E978 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 60 7E AE 60 60 6F 87 60 2ABC

E980 00 00 00 00 00 00 00 00 : 00
E988 00 00 00 00 00 00 00 00 : 00
E990 00 00 00 00 00 0F 27 20 : 56
E998 20 20 20 20 20 20 20 00 : E0
E9A0 00 00 00 00 00 00 00 00 : 00
E9A8 00 00 00 00 00 00 00 00 : 00
E9B0 00 00 00 00 00 00 00 00 : 00
E9B8 00 00 00 00 00 00 00 00 : 00
E9C0 00 0F 27 20 20 20 20 20 : D6
E9C8 20 20 20 00 00 00 00 00 : 60
E9D0 00 00 00 00 00 00 00 00 : 00
E9D8 00 00 00 00 00 00 00 00 : 00
E9E0 00 00 00 00 00 00 00 00 : 00
E9E8 00 00 00 00 00 0F 27 20 : 56
E9F0 20 20 20 20 20 20 20 00 : E0
E9F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 60 6F 87 60 60 7E AE 60 85E5

EA00 00 00 00 00 00 00 00 00 : 00
EA08 00 00 00 00 00 00 00 00 : 00
EA10 00 00 00 00 00 00 00 00 : 00
EA18 00 0F 27 20 20 20 20 20 : D6
EA20 20 20 20 00 00 00 00 00 : 60
EA28 00 00 00 00 00 00 00 00 : 00
EA30 00 00 00 00 00 00 00 00 : 00
EA38 00 00 00 00 00 00 00 00 : 00
EA40 00 00 00 00 00 0F 27 20 : 56
EA48 20 20 20 20 20 20 20 00 : E0
EA50 00 00 00 00 00 00 00 00 : 00
EA58 00 00 00 00 00 00 00 00 : 00
EA60 00 00 00 00 00 00 00 00 : 00
EA68 00 00 00 00 00 00 00 00 : 00
EA70 00 0F 27 20 20 20 20 20 : D6
EA78 20 20 20 00 00 00 00 00 : 60
---------------------------------
SUM: 60 7E AE 60 60 6F 87 60 BA5D

EA80 00 00 00 00 00 00 00 00 : 00
EA88 00 00 00 00 00 00 00 00 : 00
EA90 00 00 00 00 00 00 00 00 : 00
EA98 00 00 00 00 00 0F 27 00 : 36
EAA0 00 00 00 00 00 00 00 00 : 00
EAA8 00 00 00 00 00 00 00 00 : 00
EAB0 00 00 00 00 00 00 00 00 : 00
EAB8 00 00 00 00 00 00 00 00 : 00
EAC0 00 00 00 00 00 00 00 00 : 00
EAC8 00 0F 27 00 00 00 00 00 : 36
EAD0 00 00 00 00 00 00 00 00 : 00
EAD8 00 00 00 00 00 00 00 00 : 00
EAE0 00 00 00 00 00 00 00 00 : 00
EAE8 00 00 00 00 00 00 00 00 : 00
EAF0 00 00 00 00 00 0F 27 00 : 36
EAF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 0231

EB00 00 00 00 00 00 00 00 00 : 00
EB08 00 00 00 00 00 00 00 00 : 00
EB10 00 00 00 00 00 00 00 00 : 00
EB18 00 00 00 00 00 00 00 00 : 00
EB20 00 0F 27 00 00 00 00 00 : 36
EB28 00 00 00 00 00 00 00 00 : 00
EB30 00 00 00 00 00 00 00 00 : 00
EB38 00 00 00 00 00 00 00 00 : 00
EB40 00 00 00 00 00 00 00 00 : 00
EB48 00 00 00 00 00 0F 27 00 : 36
EB50 00 00 00 00 00 00 00 00 : 00
EB58 00 00 00 00 00 00 00 00 : 00
EB60 00 00 00 00 00 00 00 00 : 00
EB68 00 00 00 00 00 00 00 00 : 00
EB70 00 00 00 00 00 00 00 00 : 00
EB78 00 0F 27 00 00 00 00 00 : 36
---------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 5EEC

EB80 00 00 00 00 00 00 00 00 : 00
EB88 00 00 00 00 00 00 00 00 : 00
EB90 00 00 00 00 00 00 00 00 : 00
EB98 00 00 00 00 00 00 00 00 : 00
EBA0 00 00 00 00 00 0F 27 00 : 36
EBA8 00 00 00 00 00 00 00 00 : 00
EBB0 00 00 00 00 00 00 00 00 : 00
EBB8 00 00 00 00 00 00 00 00 : 00
EBC0 00 00 00 00 00 00 00 00 : 00
EBC8 00 00 00 00 00 00 00 00 : 00
EBD0 00 0F 27 00 00 00 00 00 : 36
EBD8 00 00 00 00 00 00 00 00 : 00
EBE0 00 00 00 00 00 00 00 00 : 00
EBE8 00 00 00 00 00 00 00 00 : 00
EBF0 00 00 00 00 00 00 00 00 : 00
EBF8 00 00 00 00 00 0F 27 00 : 36
---------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 268C

EC00 00 00 00 00 00 00 00 00 : 00
EC08 00 00 00 00 00 00 00 00 : 00
EC10 00 00 00 00 00 00 00 00 : 00
EC18 00 00 00 00 00 00 00 00 : 00
EC20 00 00 00 00 00 00 00 00 : 00
EC28 00 0F 27 00 00 00 00 00 : 36
EC30 00 00 00 00 00 00 00 00 : 00
EC38 00 00 00 00 00 00 00 00 : 00
EC40 00 00 00 00 00 00 00 00 : 00
EC48 00 00 00 00 00 00 00 00 : 00
EC50 00 00 00 00 00 0F 27 00 : 36
EC58 00 00 00 00 00 00 00 00 : 00
EC60 00 00 00 00 00 00 00 00 : 00
EC68 00 00 00 00 00 00 00 00 : 00
EC70 00 00 00 00 00 00 00 00 : 00
EC78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 0F 27 00 00 0F 27 00 124E

EC80 00 0F 27 00 00 00 00 00 : 36
EC88 00 00 00 00 00 00 00 00 : 00
EC90 00 00 00 00 00 00 00 00 : 00
EC98 00 00 00 00 00 00 00 00 : 00
ECA0 00 00 00 00 00 00 00 00 : 00
ECA8 00 00 00 00 00 0F 27 00 : 36
ECB0 00 00 00 00 00 00 00 00 : 00
ECB8 00 00 00 00 00 00 00 00 : 00
ECC0 00 00 00 00 00 00 00 00 : 00
ECC8 00 00 00 00 00 00 00 00 : 00
ECD0 00 00 00 00 00 00 00 00 : 00
ECD8 00 0F 27 00 00 00 00 00 : 36
ECE0 00 00 00 00 00 00 00 00 : 00
ECE8 00 00 00 00 00 00 00 00 : 00
ECF0 00 00 00 00 00 00 00 00 : 00
ECF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 968D

ED00 00 00 00 00 00 0F 27 00 : 36
ED08 00 00 00 00 00 00 00 00 : 00
ED10 00 00 00 00 00 00 00 00 : 00
ED18 00 00 00 00 00 00 00 00 : 00
ED20 00 00 00 00 00 00 00 00 : 00
ED28 00 00 00 00 00 00 00 00 : 00
ED30 00 0F 27 00 00 00 00 00 : 36
ED38 00 00 00 00 00 00 00 00 : 00
ED40 00 00 00 00 00 00 00 00 : 00
ED48 00 00 00 00 00 00 00 00 : 00
ED50 00 00 00 00 00 00 00 00 : 00
ED58 00 00 00 00 00 0F 27 00 : 36
ED60 00 00 00 00 00 00 00 00 : 00
ED68 00 00 00 00 00 00 00 00 : 00
ED70 00 00 00 00 00 00 00 00 : 00
ED78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 5B97

ED80 00 00 00 00 00 00 00 00 : 00
ED88 00 0F 27 00 00 00 00 00 : 36
ED90 00 00 00 00 00 00 00 00 : 00
ED98 00 00 00 00 00 00 00 00 : 00
EDA0 00 00 00 00 00 00 00 00 : 00
EDA8 00 00 00 00 00 00 00 00 : 00
EDB0 00 00 00 00 00 0F 27 00 : 36
EDB8 00 00 00 00 00 00 00 00 : 00
EDC0 00 00 00 00 00 00 00 00 : 00
EDC8 00 00 00 00 00 00 00 00 : 00
EDD0 00 00 00 00 00 00 00 00 : 00
EDD8 00 00 00 00 00 00 00 00 : 00
EDE0 00 0F 27 00 00 00 00 00 : 36
EDE8 00 00 00 00 00 00 00 00 : 00
EDF0 00 00 00 00 00 00 00 00 : 00
EDF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 E59E

EE00 00 00 00 00 00 00 00 00 : 00
EE08 00 00 00 00 00 0F 27 00 : 36
EE10 00 00 00 00 00 00 00 00 : 00
EE18 00 00 00 00 00 00 00 00 : 00
EE20 00 00 00 00 00 00 00 00 : 00
EE28 00 00 00 00 00 00 00 00 : 00
EE30 00 00 00 00 00 00 00 00 : 00
EE38 00 0F 27 00 00 00 00 00 : 36
EE40 00 00 00 00 00 00 00 00 : 00
EE48 00 00 00 00 00 00 00 00 : 00
EE50 00 00 00 00 00 00 00 00 : 00
EE58 00 00 00 00 00 00 00 00 : 00
EE60 00 00 00 00 00 0F 27 00 : 36
EE68 00 00 00 00 00 00 00 00 : 00
EE70 00 00 00 00 00 00 00 00 : 00
EE78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 3D0B

EE80 00 00 00 00 00 00 00 00 : 00
EE88 00 00 00 00 00 00 00 00 : 00
EE90 00 0F 27 00 00 00 00 00 : 36
EE98 00 00 00 00 00 00 00 00 : 00
EEA0 00 00 00 00 00 00 00 00 : 00
EEA8 00 00 00 00 00 00 00 00 : 00
EEB0 00 00 00 00 00 00 00 00 : 00
```

▶劇場用アニメ「はじまりの冒険者たち～レジェンド・オブ・クリスタニア～」のデキは尋常じゃない。レンタルが出たら，ぜひ借りましょう。　河村　憲昭(21)愛知県

```
EEB8 00 00 00 00 00 0F 27 00 : 36
EEC0 00 00 00 00 00 00 00 00 : 00
EEC8 00 00 00 00 00 00 00 00 : 00
EED0 00 00 00 00 00 00 00 00 : 00
EED8 00 00 00 00 00 00 00 00 : 00
EEE0 00 00 00 00 00 00 00 00 : 00
EEE8 00 0F 27 00 00 00 00 00 : 36
EEF0 00 00 00 00 00 00 00 00 : 00
EEF8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 21E4

EF00 00 00 00 00 00 00 00 00 : 00
EF08 00 00 00 00 00 00 00 00 : 00
EF10 00 00 00 00 00 0F 27 00 : 36
EF18 00 00 00 00 00 00 00 00 : 00
EF20 00 00 00 00 00 00 00 00 : 00
EF28 00 00 00 00 00 00 00 00 : 00
EF30 00 00 00 00 00 00 00 00 : 00
EF38 00 00 00 00 00 00 00 00 : 00
EF40 00 0F 27 00 00 00 00 00 : 36
EF48 00 00 00 00 00 00 00 00 : 00
EF50 00 00 00 00 00 00 00 00 : 00
EF58 00 00 00 00 00 00 00 00 : 00
EF60 00 00 00 00 00 00 00 00 : 00
EF68 00 00 00 00 00 0F 27 00 : 36
EF70 00 00 00 00 00 00 00 00 : 00
EF78 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 6E4A

EF80 00 00 00 00 00 00 00 00 : 00
EF88 00 00 00 00 00 00 00 00 : 00
EF90 00 00 00 00 00 00 00 00 : 00
EF98 00 0F 27 00 00 00 00 00 : 36
EFA0 00 00 00 00 00 00 00 00 : 00
EFA8 00 00 00 00 00 00 00 00 : 00
EFB0 00 00 00 00 00 00 00 00 : 00
EFB8 00 00 00 00 00 00 00 00 : 00
EFC0 00 00 00 00 00 0F 27 00 : 36
EFC8 00 00 00 00 00 00 00 00 : 00
EFD0 00 00 00 00 00 00 00 00 : 00
EFD8 00 00 00 00 00 00 00 00 : 00
EFE0 00 00 00 00 00 00 00 00 : 00
EFE8 00 00 00 00 00 00 00 00 : 00
EFF0 00 0F 27 00 00 00 00 00 : 36
EFF8 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 1FFD

F000 00 00 00 00 00 00 00 00 : 00
F008 00 00 00 00 00 00 00 00 : 00
F010 00 00 00 00 00 00 00 00 : 00
F018 00 00 00 00 00 0F 27 00 : 36
F020 00 00 00 00 00 00 00 00 : 00
F028 00 00 00 00 00 00 00 00 : 00
F030 00 00 00 00 00 00 00 00 : 00
F038 00 00 00 00 00 00 00 00 : 00
F040 00 00 00 00 00 00 00 00 : 00
F048 00 0F 27 00 00 00 00 00 : 36
F050 00 00 00 00 00 00 00 00 : 00
F058 00 00 00 00 00 00 00 00 : 00
F060 00 00 00 00 00 00 00 00 : 00
F068 00 00 00 00 00 00 00 00 : 00
F070 00 00 00 00 00 0F 27 00 : 36
F078 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 00 0F 27 00 00 1E 4E 00 0231

F080 00 00 00 00 00 00 00 00 : 00
F088 00 00 00 00 00 00 00 00 : 00
F090 00 00 00 00 00 00 00 00 : 00
F098 00 00 00 00 00 00 00 00 : 00
F0A0 00 0F 27 00 00 00 00 00 : 36
F0A8 00 00 00 00 00 00 00 00 : 00
F0B0 00 00 00 00 00 00 00 00 : 00
F0B8 00 00 00 00 00 00 00 00 : 00
F0C0 00 00 00 00 00 00 00 00 : 00
F0C8 00 00 00 00 00 0F 27 00 : 36
F0D0 00 00 00 00 00 00 00 00 : 00
F0D8 00 00 00 00 00 00 00 00 : 00
F0E0 00 00 00 00 00 00 00 00 : 00
F0E8 00 00 00 00 00 00 00 00 : 00
F0F0 00 00 00 00 00 00 00 00 : 00
F0F8 00 0F 27 00 00 00 00 00 : 36
----------------------------------
SUM: 00 1E 4E 00 00 0F 27 00 5EEC

F100 00 00 00 00 00 00 00 00 : 00
F108 00 00 00 00 00 00 00 00 : 00
F110 00 00 00 00 00 00 00 00 : 00
F118 00 00 00 00 00 00 00 00 : 00
F120 00 00 00 00 00 0F 27 00 : 36
F128 00 00 00 00 00 00 00 00 : 00
F130 00 00 00 00 00 00 00 00 : 00
F138 00 00 00 00 00 00 00 00 : 00
F140 00 00 00 00 00 00 00 00 : 00
F148 00 00 00 00 00 00 00 00 : 00
F150 00 0F 27 20 20 20 20 20 : D6
F158 20 20 20 00 00          : 60
----------------------------------
SUM: 20 2F 47 20 20 2F 47 20 DF25
```

リスト3

```
//*******************************************/
//*                                         */
//*  P I C T   P U Z Z L E用  データクリア  */
//*                                         */
//*    要S－OS  SLANG                       */
//*                                         */
//*                    Presented By BCMA    */
//*******************************************/
MAIN()
VAR I,J;
ARRAY BYTE DAT[21]=[$50,$49,$43,$54,$50,$54,$5A,$5A,
$4C,$C5,$00,$44,$41,$54,$41,$00,$42,$59,$42,$43,$4D,$41 ];
BEGIN
I=$E000;
PRINT("NOW_REFRESH!");
FOR J=0 TO 44*101[
MEM[I+J]=0;
]
FOR J=0 TO 101[
MEM[I+J*44+32]=0;
MEMW[I+J*44+33]=9999;
]
FOR J=0 TO 21[
MEM[I+J]=DAT[J];
]
WIDTH(40);
END;
```

▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは、基本システムであるS-OS "MACE" またはS-OS "SWORD" がないと作動しませんのでご注意ください。

1987

■87年1月号
第35部 マシン語入力ツールMACINTO-C
連載 FuzzyBASIC料理法<4>
■87年2月号
第36部 アドベンチャーゲームMARMALADE
第37部 テキアペ作成ツールCONTEX
■87年3月号
第38部 魔法使いはアニメがお好き
第39部 アニメーションツールMAGE
付録 “SWORD”再掲載とMAGICの標準化
■87年4月号
第40部 INVADER GAME
第41部 TANGERINE
■87年5月号
第42部 S-OS“SWORD”変身セット
第43部 MZ-700“SWORD”をQD対応に
■87年6月号
インタラプト コンパイラ物語
第44部 FuzzyBASICコンパイラ
第45部 エディタアセンブラZEDA-3
■87年7月号
第46部 STORY MASTER
■87年8月号
第47部 パズルゲーム碁石拾い
第48部 漢字出力パッケージJACKWRITE
特別付録 FM-7/77版S-OS“SWORD”
■87年9月号
第49部 リロケータブル逆アセンブラInside-R
特別付録 PC-8001/8801版S-OS“SWORD”
■87年10月号
第50部 tiny CORE WARS
第51部 FuzzyBASICコンパイラの拡張
第52部 XIturob版S-OS“SWORD”
■87年11月号
序論 神話のなかのマイクロコンピュータ
付録 S-OSの仲間たち
第53部 もうひとつのFuzzyBASIC入門
第54部 ファイルアロケータ＆ローダ
インタラプト S-OSこちら集中治療室
第55部 BACK GAMMON
■87年12月号
第56部 タートルグラフィックパッケージTURTLE
第57部 XIturbo版“SWORD”アフターケア
ラインプリントルーチン
特別付録 PASOPIA7版S-OS“SWORD”

1988

■88年1月号
第58部 FuzzyBASICコンパイラ・奥村版
付録 石上版コンパイラ拡張部の修正
■88年2月号
第59部 シューティングゲームELFES
■88年3月号
第60部 構造型コンパイラ言語SLANG
■88年4月号
第61部 デバッギングツールTRADE
第62部 シミュレーションウォーゲームWALRUS
■88年5月号
第63部 シューティングゲームELFES II
第64部 地底最大の作戦
■88年6月号
第65部 構造化言語SLANG入門(1)
第66部 Lisp-85用NAMPAシミュレーション
■88年7月号
第67部 マルチウィンドウドライバーMW-1
連載 構造化言語SLANG入門(2)
■88年8月号
第68部 マルチウィンドウエディタWINER
■88年9月号
第69部 超小型エディタTED-750
第70部 アフターケアWINERの拡張
■88年10月号
第71部 SLANG用ファイル入出力ライブラリ
第72部 シューティングゲームMANKAI
■88年11月号
第73部 シューティングゲームFLFESIV
■88年12月号
第74部 ソースジェネレータSOURCERY

1989

■89年1月号
第75部 パズルゲームLAST ONE
第76部 ブロックゲームFLICK
■89年2月号
第77部 高速エディタアセンブラREDA
特別付録 XI版S-OS“SWORD”<再掲載>
■89年3月号
第78部 Z80用浮動小数点演算パッケージSOR
OBAN
■89年4月号
第79部 SLANG用実数演算ライブラリ
■89年5月号
第80部 ソースジェネレータRING
■89年6月号
第81部 超小型コンパイラTTC
■89年7月号
第82部 TTC用パズルゲームTICBAN
■89年8月号
第83部 CP/M用ファイルコンバータ
■89年9月号
第84部 生物進化シミュレーションBUGS
■89年10月号
第85部 小型インタプリタ言語TTI
■89年11月号
第86部 TTI用パズルゲームPUSH BON!
■89年12月号
第87部 SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部 SLANG用ゲームWORM KUN
特別付録 再掲載SLANGコンパイラ
■90年2月号
第89部 超小型コンパイラTTC＋＋
■90年3月号
第90部 超多機能アセンブラOHM-Z80
■90年4月号
第91部 ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部 インタプリタ言語STACK
■90年6月号
第93部 リロケータブルフォーマットの取り決め
第94部 STACK用ゲームSQUASH!
第95部 X68000対応S-OS“SWORD”
特別付録 PC-286対応S-OS“SWORD”
■90年7月号
第96部 リロケータブルアセンブラWZD
■90年8月号
第97部 リンカWLK
■90年9月号
第98部 BILLIARDS
■90年10月号
第99部 ライブラリアンWLB
■90年11月号
第100部 タブコート対応エディタEDC-T
■90年12月号
第101部 STACKコンパイラ

1991

■91年1月号
第102部 ブロックアクションゲームCOLUMNS
■91年2月号
第103部 ダイスゲームKISMET
■91年3月号
第104部 アクションゲームMUD BALLIN'
■91年4月号
第105部 SLANG用カードゲームDOBON
■91年5月号
第106部 実数型コンパイラ言語REAL
■91年6月号
第107部 Small-C処理系の移植
■91年7月号
第108部 REALソースリスト編
■91年8月号
第109部 Small-Cライブラリの移植
■91年9月号
第110部 SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部 Small-C活用講座（初級編）
■91年11月号
第112部 Small-C活用講座（応用編）
第113部 MORTAL
■91年12月号
第114部 Small-C SLANGコンパチ関数

1992

■92年1月号
第115部 LINER
■92年2月号
第116部 シミュレーションゲームPOLANYI
■92年3月号
第117部 カードゲームKLONDIKE
■92年4月号
第118部 オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部 COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部 COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部 関数リファレンス実践Small-C講座(4)
■92年8月号
第122部 ワイルドカード実践Small-C講座(5)
第123部 グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部 O-EDIT&MODCNV
■92年10月号
第125部 SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部 EDIT実線Small-C講座(7)
■92年12月号
第127部 MAKE実践Small-C講座(8)

1993

■93年1月号
第128部 EDC-Tの拡張
■93年2月号
第129部 BLACK JACK
■93年3月号
第130部 シューティングゲームコアシステム作成法(1)
■93年4月号
第131部 シューティングゲームコアシステム作成法(2)
■93年5月号
第132部 シューティングゲームコアシステム作成法(3)
■93年6月号
第133部 REVERSI
■93年7月号
特別付録 MSX用S-OS“SWORD”
■93年8月号
第134部 MACINTO-C再掲載
■93年9月号
第135部 7並べ
特別付録 SLANG再々掲載
■93年10月号
第136部 シューティングゲームコアシステム作成法(4)
■93年11月号
第137部 S-OSで学ぶZ80マシン語講座(1)
■93年12月号
第138部 エディタアセンブラREDA再掲載

1994

■94年1月号
第139部 S-OSで学ぶZ80マシン語講座(2)
■94年2月号
第140部 YGCSver.0.20ユーザーマニュアル
第141部 S-OSで学ぶZ80マシン語講座(3)
■94年3月号
第142部 S-OSで学ぶZ80マシン語講座(4)
■94年4月号
第143部 S-OSで学ぶZ80マシン語講座(5)
■94年5月号
第144部 S-OSで学ぶZ80マシン語講座(6)
■94年6月号
第145部 YGCSver.0.30
■94年7月号
第146部 シューティングゲーム作成講座(1)
■94年8月号
第147部 シューティングゲーム作成講座(2)
■94年9月号
第148部 怪しいZ80の使い方(テクニック編)
■94年10月号
第149部 シューティングゲーム作成講座(3)
第150部 怪しいZ80の使い方(未定義命令編)
■94年11月号
第151部 B-GALET2
■94年12月号
第152部 シューティングゲーム作成講座(4)

シャーペンワープロパック ver.2.0

Taki Yasushi 瀧 康史

最初に断言します。

「シャーペンワープロパック ver.2.0」(以下ワープロパック2)を筆頭に,使えるフォント群*1を集めてくると,かなり表現力があるワープロシステムができ上がります。操作性に関して,多少謎な部分はありますが,カスタマイズの自由度を考慮すると,かなりよくできたワープロです。

意味不明な機能が徐々に増えている昨今のワープロのなかで,ワープロパック2の構成は,かなりスマートです。基本的に高性能エディタ指向ではありますが,マルチフォントに関する部分などのワープロ系機能はマウスに振り分けられているからです。このカスタマイズに疑問を感じる人は,キーからメニュー,そしてワードラッピングなどの文字まで,カスタマイズを自由に行うことができます。

そのワープロパック2を利用するためにはどうすればいいか。それは,前回のワープロパック1と同じく,SX-WINDOW ver.3.1システムが必要になります。結果的にワープロパック2はSX-WINDOW(の内部リソース)の印刷機能のバージョンアップや,追加なども行います。

1 フォント&ロゴツール集,書体クラブ,JGワードなどより,拡張子が.ifm, *.vf?, *.fn?のもの。

拡張されるシステムと新規対応ハード群

ワープロパック2は,基本的にSX-WINDOW Ver3.1に標準添付されたシャーペン.Xを拡張する形をとっています。SX-WINDOWのシステムのバージョンアップに関しては,

・色変換モジュールのバージョンアップ
・カラーインクジェットプリンタに対応(MJ-700V2C,MJ-800Cのマッハジェット系列,MJ-900C,MJ-5000Cのエスパーマッハ系列,BJC-400J,BJC-600Jのバブルジェット系列)

などが挙げられます。

主に拡張されたのは印刷に関する部分です。SX-WINDOWのシステムに関する部分は,当然ながらSX-WINDOWで動くアプリケーションすべてに影響します。具体的に恩恵が得られるのは,XDTP,Easydrawなどでしょうか。

カラープリンタへの対応に関しては,10月号のカラーページに掲載したとおり,なかなかの印刷を行います。MJ-500Cは基本的には未対応ですが,360dpiモードで利用できるはずです(ただしMJ-500Cがないので未確認)。具体的には,従来からあったESC/P24-J84・Cに,新しく追加される色変換リソースである「誤差拡散」を設定さえすれば,よいはずです。セミ720dpiでは印字できませんが,360dpiでBJC-400J程度には印刷できるでしょう。また,おまけとしてMJシリーズ専用のギャップ調整プログラムがついています。ほかにもCD-ROMデバイスドライバ「CDDEV.SYS」の機能限定版が付属するなど,盛りだくさんです。

ワープロとしての新機能

SX-WINDOW ver.3.1に付属したシャーペンから拡張された機能としては(以下,マニュアルより抜粋),

1) レイアウトモード
2) マスターフォーム
3) ノンブル,日付,時刻のスタンプ
4) 両面印刷,袋綴じ印刷
5) 段組み編集
6) スタイルパレット
7) ブック構成
8) CD-ROM辞書検索機能
9) 縦書き入力
10) 禁則処理の拡張
11) 行揃えの拡張
12) ルーラの表示
13) タブ/インデントの設定
14) 文字の回転
15) 特殊ペースト
16) マクロ機能の拡張
17) 強制改行幅0の導入
18) 禁則処理,行揃え,タブの設定がパラグラフ単位で設定可能
19) 行揃えの幅の計算に改行コードが含まれない

などが挙げられます。見てわかるとおり,どちらかというと,印刷に「こだわる」人向けの機能が主に追加されているので,ワープロパック1ユーザーであり,印刷機能に特に不満がないのであるなら,新しい機能はいらないかもしれません(逆に機能が増えると使いづらくなる人もいるためです)。ただし,先に書いたとおり,SXシステムそのものを拡張できたり,開発者向けのライブラリ,リファレンスなどが入っていることを考えると,買っておくべきでしょう。

それではバージョンアップで追加された機能について説明していきましょう。ただし,9)~19)の機能についてはワープロパック1から対応していますので,1995年2月号のレビューを参考にしてください。

ワープロパック1では,WYSIWYG(What you see is what you get)をうたうもの

図1 印字イメージがつかめるレイアウト表示

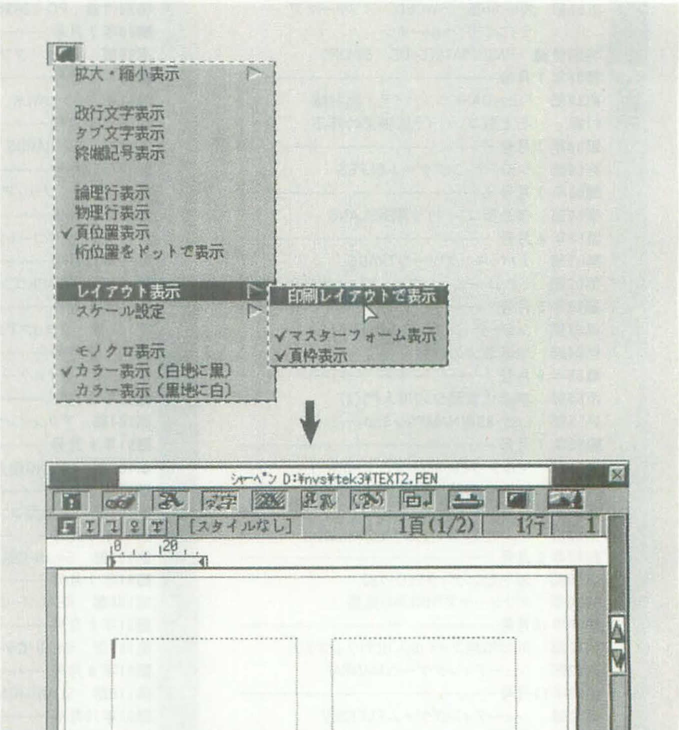
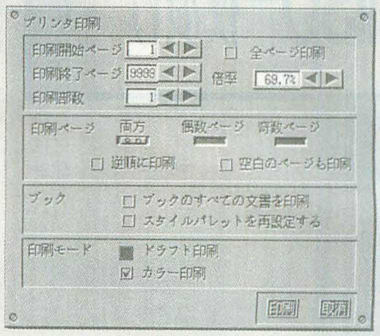


図2　イメージ印刷時のダイアログ



の，多段落などを利用すると，実際に紙に書かれた状態を簡単に想像できませんでした。1)のレイアウトモードでは，特に「紙」をイメージしているため，実際に紙にどうやって印刷されるかがわかるようになっています(図1)。ただ，このモードでは編集処理が少し重くなります。

2)のマスターフォームは去年の私のXDTPの連載で，細かく話しましたが，これとは少し違います。あちらは，DTPらしく，ドキュメントの根幹となるものですが，こちらは，ワープロらしく，あくまでも文章ファイルの雛形になるだけです。

たとえば，いままでシャーペンを使っていて，環境をいくつかもちたいと思ったことはありませんか？ カセットレーベルを印刷するときの環境，A4レポートを印刷するときの環境，ハガキを印刷したいときの環境，リストを印刷したいときの環境など，それらの環境を雛形として管理することができるということです。

3)のノンブルとはページ番号のこと。せっかく綺麗に印刷したのに，シャーペンで出力すると，「ページ番号が印刷されない！」と思ったことはあるでしょう？

4)はブック構成と組み合わせて使えば，印刷のバリエーションが広がります。

5)の段組みは印刷メニューのページレイアウトで設定できます。ワープロパック1では印刷時のみの設定でしたが，文書作成段階から確認できるようになりました。

6)のスタイルパレットは，フォント名，フォントサイズ，文字飾りなどの文字単位の情報，タブインデントのパラグラフ情報などを，まとめて名前をつけて保存するモードです。

いままではこれができないため，見出し，小見出し，強調などの設定は，いつも同じことをするのにもかかわらず，1カ所ずつ設定してきました。たとえば私の環境は，見出しは28ドット/JGゴシック，小見出しは22ドット/JG飛鳥などと決まっています。これを名前として登録しておくことにより，違う環境で「小見出し」「大見出し」などを一気に印刷することが可能です。これと，マスターフォームを利用すると，かなり環境に依存しないで文書を書くことができます。たとえば，ネットワークのフリーソフトのマニュアルなど，この機能を使えば，スタイルパレット/マスターフォームを変えるだけで，ずいぶん綺麗に印刷できることになるのです。

7)のブック構成というのは，複数のpenファイルを，ひとつの塊としてまとめておくような感じです。ただしこれは，ディスクイメージ上ではなくて，印刷時の便宜上の問題と考えればいいと思います。まとめて印刷したいときなどに，ブック構成と設定して印刷すると，まとめて印刷できるというわけです。この機能とマスターフォーム，スタイルパレットによって，ワープロパック2のマニュアルは，簡単に印刷できるようになったのです。

その他……

CD-ROM検索ソフトなどは，うれしい人には猛烈にうれしいかもしれません。個人的には，印刷時に90度回転など，もう少し縦書きに重点をおいてほしかったと思いますが，ドキュメントを印刷しあうことができるのが，いいかなと思いました。

スタイルに関しては，マニュアルにいくつか登録してありますから，コピーすればいいでしょう。ただ，もう少し充実しているとよかったのですが。

マスターフォームに関してもそうです。たとえば，エーワンラベルの型番○○対応とか，カセットインデックス対応とか，ハガキ対応とか，そのあたりのものはそれだけでも価値があったかもしれません。まあ，今後，ネットなどでお互いにアップされまくるのかな？

個人的には，シャーペンの印刷系で最も，利用しづらい「ドットでものを測る」という概念をなんとかするモードがあればよかったかなと思いました。というのは，標準的なWindowsワープロは，画面に表示されている16ドットフォントが，12ポイント文字として印刷されます。ですから，「16ドットフォント」を「12ポイント」で印刷とか，「8級」で印刷とか指定できるモードがあれば，わかりやすかったのではないかと思います(図2)。

結果的に倍率を自分で計算すればできるので，カスタマイズすれば問題ありませんが，標準でこういったものが入るのと入らないのとでは，使い勝手が全然違うと思います(コラム参照)。

同じような感じで，ワープロパック2のキーカスタマイズはあくまでもエディタ指向です。ワープロ指向のカスタマイズもおまけでついていてもいいかなと思いました。

今回はワープロパック2のワープロとしての進化を中心に紹介してきました。最初にも書きましたが，印刷機能を強化したい人は買いでしょう。それに，ほかにも外部コマンドやライブラリも充実しているのですが，このあたりの話は12月号でけんと君におまかせということで。

シャーペンワープロパック ver.2.0　9,800円(税別)
計測技研　☎0286(22)9811

思った文字の大きさで印刷できない人へ

「画面サイズ」で印刷にすると，妙に16ドットフォントが大きくなるし，「そのまま」のサイズで印刷すると，むちゃくちゃ小さくなってしまいます。実は画面サイズというのは，15インチモニタで768×512ドットで印刷したときに，画面のレイアウトがA4サイズになるということなんですね。そのままというのは，画面上のドットがプリンタの1ドットになっているというわけです。

でも，画面の16ドットフォント文字は，プリンタでは「あの」サイズになってほしいというのがあるじゃないですか。「あの」サイズというのはコード印字のサイズです。たとえば360dpiプリンタの場合，コード印字のサイズは，48ドットですから，16ドットフォントを48ドットで印刷するためには，48/16＝3倍。つまり，そのままのサイズで印刷するためには，300%で印刷すれば，ちょうど16ドットフォントがコード印字の文字サイズになります。倍率をあらかじめ決めておくと，ある程度思ったとおりに印刷することができます。

リストとはなにか

Tamura Kento 田村 健人

リストはLispが扱う本質的なデータ構造だ
Lispを学ぶ際にリスト構造を理解することは不可欠なことである
今回はリストの基本的な概念とリスト処理関数を見てみよう

やはりどこか悪くするとひしひしと健康のありがたみを感じる。8月中に腰がおかしくなって，腰を曲げている状態で静止することができなくなってしまった。座っていられないのである。前回の原稿はディスプレイを床に置いて，寝転がりながら書いた。1カ月経ったいまでは普段の生活に支障がない程度にはなったが，長時間座っているとつらい。

運動不足や姿勢の悪さが原因なんだろうか。ともかく健康には気をつけたいものである。

Lispにおけるデータ型

Lispで扱うデータ型には以下のようなものがある。

```
┌ アトム ──┬ 数値
│          ├ 文字列
│          ├ シンボル
│          └ nil
└ リスト
```

数値などのように，データ型としてそれ以上分解できないものをアトム(atom「原子」の意)という。アトムではないものをリストという。リストはアトムまたはリストを束ねたものである。リストを分解していくとアトムになる。

アトムの種類は処理系による。上に挙げたものはほぼすべての処理系に存在するだろう。

シンボル以外のアトムを評価するとそれ自身が評価結果となる。前回，「数値345を評価すると345が結果となる」ということを述べた。文字列も同様で，文字列を評価するとそれ自身が結果となる。シンボルは変数名や関数名に使われるデータで，シンボルを評価すると変数として代入された値を返す。

リストにも種類があって，「純リスト」と「純リストでないリスト」に分けられる。

ほかのプログラミング言語での配列とLispでのリストを較べてみよう。あるひとつの型のデータを並べることができる点は共通である。しかしリストは単一のデータ型だけではなく，どんな型でも並べることができる。

C言語などの構造体とリストを較べてみる。いろいろな型のデータを組み合わせてひとつのデータとして扱うことができる点は共通である。しかしリストでは各要素の型を制限することはないし，動的に構造を変化させることができる。

リストは配列の代わりにもなるし，構造体の代わりにもなるのである。なお，Common Lispではリストとは関係なく配列と構造体を言語仕様に含めている。

ドット対

リストを作るためのもっとも基本的な関数がconsである。

```
(cons 123 456)
```

の評価結果は，

```
(123 . 456)
```

となる。括弧の中に，第1引数の評価結果，ピリオド，第2引数の評価結果が書かれている。この形式のデータのことを「ドット対(dotted pair)」と呼ぶ。ドット対はアトムではないので，リストである。ドット対の，ピリオドの左側を「car(かぁ)」部，右側を「cdr(くだぁ)」部と呼ぶ。ドット対に入っているデータを取り出す関数がcarとcdrである。carは引数に与えられたドット対のcar部を返し，cdrはcdr部を返す。

```
(car (cons 123 456))  →  123
(cdr (cons 123 456))  →  456
```

consの動作について解説しよう(図1)。まず2つの引数が評価される。評価結果はどこかに置いておく。次にconsセルが生成される。consセルは2本の矢印を持っていて，それぞれがcar部，cdr部に対応している。そして，car部の矢印で第1引数の評価結果，cdr部の矢印で第2引数の評価結果を指す。この状態でconsセルを評価結果とするのである。なお，consセルを図示するときは当然car部が左側，cdr部が右側に描かれる。

次に，ドット対を変数に代入してみよう。

(setq hoge (cons 123 456))

いちいちconsを使うのは面倒なので，ドット対を直接代入したい。

(setq hoge (123 . 456))

これは間違いである。Lispインタプリタは「(123 . 456)」を評価しようとするが，アトムではないし関数呼び出しでもないのでエラーとなってしまうのである。そこで，「評価しないでそのまま扱って」と指令する必要がある。引数を評価しないでそのまま返すquoteという特殊形式がある。

(quote (123 . 456))

を評価すると，

(123 . 456)

である。これを用いれば，

(setq hoge (quote (123 . 456)))

と書けるのである。しかし，これでも冗長であるので，quoteには省略記法がある。

(quote A)

は，

'A

と書くことができる。結局，次のようにすればよい。

(setq hoge '(123 . 456))

ドット対からリストへ

ドット対はリストの基本ではあるが，データを2つしか持てないように見える。実際にはドット対の中にドット対を入れることができるので，いくらでもデータを入れることができる。

(setq month '(("睦月" . "如月") . ("弥生" . "卯月")))　(図2)

ドット対を使って4つのデータを格納してみた。このリストのそれぞれの要素にアクセスするためには，以下のようにすればよい。

"睦月" ← (car (car month))
"如月" ← (cdr (car month))
"弥生" ← (car (cdr month))
"卯月" ← (cdr (cdr month))

この例では各要素は同列に扱いたいとしよう。しかし上記の構造ではほしい要素ごとにアクセス方法を変えなければならず，不便である。

そこで，次のような構造にする。

(setq month2 '("睦月" . ("如月" . ("弥生" . ("卯月" . nil)))))　(図3)

各ドット対のcar部に要素を置き，cdr部に次のドット対を置いている。最後の要素があるドット対のcdr部はnilとする。各要素のアクセス方法は次のようになる。

"睦月" ← (car month2)
"如月" ← (car (cdr month2))
"弥生" ← (car (cdr (cdr month2)))
"卯月" ← (car (cdr (cdr (cdr month2))))

単純にcdrする回数が増えるだけだということがわかるだろう。

この形式のリストのことを「純リスト(pure list)」と呼ぶ。純リストは，

・carすると先頭の要素を返す
・cdrすると先頭の要素を除いた純リストが返る

という性質を持っている。純リストは簡単に表記することができる。

(setq month2 '("睦月" . ("如月" . ("弥生" .

図1　CONSの動作

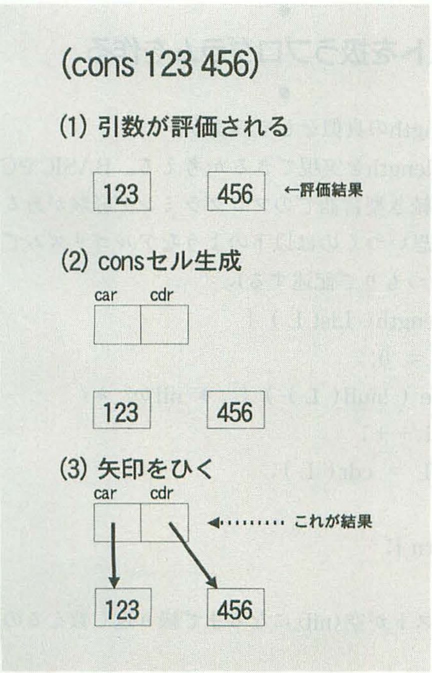


図2　入れ子になったドット対

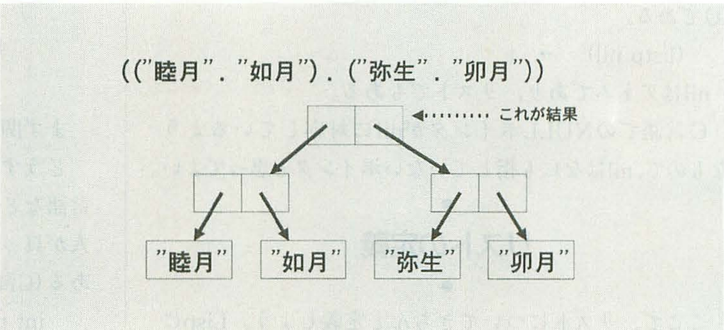
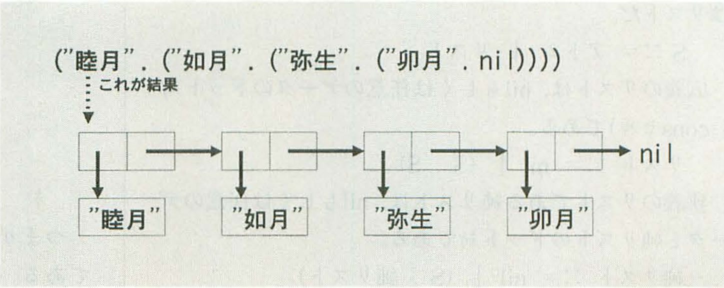


図3　純リスト形式



("卯月" . nil)))))

は,

(setq month2 '("睦月" "如月" "弥生" "卯月"))

と同じ意味である。純リストのこの表記は各データがフラットに置かれているように見えるので,「"睦月"が1番目の要素で,"如月"が2番目の要素で……」といいやすい。もちろん配列とは異なって実際にはドット対によるツリー構造をしているのである。

リストをリストの要素にしても構わないし,純リストでないリストと純リストを交ぜても構わない。

Lisp中で「リスト」というと純リストのことを指していることが多い。純リストではないリストよりは純リストのほうがいろいろな意味で扱いやすくなっているので,特別な理由がない限りは純リストを使うようにしよう。

nilについて

前回,nilを「偽を表す値」として紹介した。このnilがリストの終端を表す用途に使われている。nilはこれ以上分解できないデータであるから,アトムかどうかを判定する述語関数atomに渡した結果は真(t)である。

(atom nil) → t

要素がひとつしかない純リストにcdrするとなにが返るのだろうか。

(cdr '("卯月"))

これは,

(cdr '("卯月" . nil))

と同じ意味であるから,結果はnilである。「純リストをcdrすると先頭の要素を除いた純リストが返る」より,nilというのは要素数0の純リストであるといえる。リストかどうかを判定する述語関数listpにnilを与えた結果は真(t)である。

(listp nil) → t

nilはアトムであり,リストでもある。

C言語でのNULLポインタがnilに対応しているようなもので,nilはなにも指していないポインタと思ってよい。

リストの定義

ここで,リストについてきちんと定義しよう。Lispにおいて扱えるすべてのデータをSとする。Sはアトムまたはリストだ。

S ::= アトム | リスト

広義のリストは,nilもしくは任意のデータのドット対(=consセル)である。

リスト ::= nil | (S . S)

狭義のリストである純リストは,nilもしくは任意のデータと純リストのドット対である。

純リスト ::= nil | (S . 純リスト)

リストを扱う関数

リストを扱うための基本的な関数についていくつか解説する。詳しいことは各処理系のマニュアルを見てほしい。

●(car LIST) (cdr LIST)

先に説明したとおり,ドット対のcar部とcdr部を返す。引数が純リストならば,carは先頭の要素を取り出すという意味で,cdrは2番目以降の要素からなるリストを返すという意味である。cXXr, cXXXr, cXXXXr (Xはaまたはd)という関数もあり,たとえば,

(caddr LIST)

は,

(car (cdr (cdr LIST)))

と同じである。

●(length LIST)

リストの長さ(=要素数)を整数で返す。

(length '("睦月" "如月" "弥生" "卯月"))

は4である。

(length '("FTP" ("TCP" "UDP") "IP"))

は,3である。このリストは,2番目の要素がリストになっている。いちばん外側のリストにとってみれば中にあるリストはひとつの要素でしかないわけだ。

●(list E1 E2 E3 ……)

E1の評価結果を第1要素,E2の評価結果を第2要素,E3の評価結果を第3要素……の純リストを返す。

(list (+ 2 3) '(+ 2 3) 5)

の結果は,

(5 (+ 2 3) 5)

というリストである。

リストを扱うプログラムを作る

まず関数lengthの真似をしてみよう。

どうすればlengthを実現できるか考える。BASICやC言語などの手続き型言語でのプログラミング経験がある人が真っ先に思いつくのは以下のようなアルゴリズムである(C言語のつもりで記述する)。

```
int my_length( List L ) {
    int i = 0;
    while ( !null( L ) ) { /* nilか? */
        i ++;
        L = cdr( L );
    }
    return i;
}
```

つまり,リストが空(nil)になるまで繰り返し数えるのである。

Lispでもdoやwhileなどの繰り返しを利用できるのでこのアルゴリズムをそのまま用いることもできる。が，それはLispらしくないプログラムである。Lispらしいプログラムでは以下のように考える。

1) 空リストの長さは0である

2) あるリストの長さは，そのリストの先頭要素を取り除いたリストの長さに＋1したものである

純リストにcdrすると要素がひとつ減った純リストになるということをそのまま利用するのである。

```
(defun my-length (L)
  (cond ((null L) 0)
        (t (1+ (my-length (cdr L))))))
```

このように自分を定義するために自分自身を利用することを「再帰的定義」という。

純リスト自体が再帰的な構造をしているので，純リストを扱うプログラムも再帰アルゴリズムで記述しやすいのである。

ついでにこの関数をもとにして「リスト中のアトムの数を返す関数」を作ろう。つまり，

```
(length2 '("FTP" ("TCP" "UDP") "IP"))
```

で4を返す関数だ。

やることは簡単で，要素がリストだったら自分自身を使ってリスト中の要素を数えるのである。

```
(defun length2 (L)
  (cond ((null L) 0)
        ((listp (car L)) (+ (length2 (car L))
                            (length2 (cdr L))))
        (t (1+ (length2 (cdr L))))))
```

次に，リストを作る関数を定義してみる。たとえば，

("Solaris" "IRIX" "HP-UX" "NeXTSTEP")

のような文字列を要素とした任意長のリストから，

(7 4 5 8)

各要素の文字列の長さを要素としたリストを作る。

リストを扱う関数を作るコツは，まず空のリストを与えられた場合について考え，次にリストの長さが1減ったリストとの関係を見つけ出すことである。この場合，空のリストを与えられたら空のリストが返ればよい。長さがひとつ短いリストの結果のリストの先頭に先頭要素の文字列の長さを入れると結果となる(図4)。

```
(defun string-to-length (L)
  (cond ((null L) nil)
        (t (cons (length (car L)) (string-to-length (cdr L))))))
```

文字列の長さを返す関数はlengthだ。リストの長さを返す関数もlengthだが，リストが与えられたときはリストの長さ，文字列が与えられたときは文字列の長さを返すようになっている。consに要素とリストを渡せばリストを作ることができる。

```
(string-to-length '("NetWare" "Warp Connect" "WindowsNT"))
```

の評価結果は，

(7 12 9)

である。

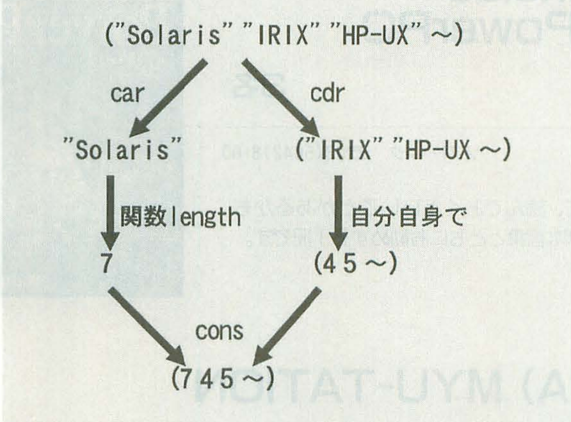
●

今回はここまで

●

Lispの基本的なプログラミングについての解説はこれで終わりにする。理解できないことや取り上げてほしいことがあったら早めにアンケートハガキなどで連絡してほしい。次回はLispの応用ということで，本来の目的(？)であるdivのカスタマイズなどについて解説する。

図4 再帰を使ったリスト処理の流れ



Coffeep?

8月号のdivの解説中で，

「Lisperはコーヒーを飲めない」

ということを書いた。これはLispにまつわる有名なパラドックスのひとつだ(単にジョークともいう)。この連載を書くにあたってLisp関係の書籍を何冊か目を通してみたわけだが，意外とこの件に関して触れているものがないのである。

Lispでは「〜かどうか」を判定する述語関数には，その「〜」に相当する単語の最後にpがついているかたちのものが多いということを前回述べた。Lispマニアたちはこれを日常生活にも応用する。ある単語の末尾にpをつけることで「〜するか？」という疑問文とするのである。たとえば，「coffeep?」は「コーヒーを飲みますか？」という意味だ。

そして，ここからが本題。

同様に，Lispの世界では，真/偽をt/nilで表すというのも前回に書いたことと思う。当然のように，Lisperたちは日常生活ではYes/Noの代わりにt/nilを使うわけだ。

では，Lispマニアが，

「Coffeep？(コーヒーを飲みますか？)」

と聞かれたときにはどう答えるのだろうか？

飲みたくなかったら「nil」と答えるだろう。これはまったく問題ない。飲みたいときは「t」と答える。しかし「t」と答えると紅茶が渡されるのである。

愛読者プレゼント

Present

プレゼントの応募方法

とじ込みのアンケートハガキの該当項目をすべてご記入のうえ，希望するプレゼント番号をハガキ右下のスペースにひとつ記入してお申し込みください。締め切りは1995年10月末日の到着分までとします。当選者の発表は1995年12月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

1 同人ソフトセット

Glossy Gloves

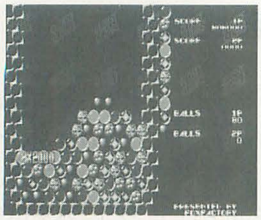
X68000用　5″/3.5″ 2HD版　500円(税込)

エリオン・クエスト

X68000用　5″/3.5″ 2HD版　500円(税込)

3名

TAKERU事務局　☎052(824)2493





対戦可能な落ちものパズルとテーブルトークRPG風アドベンチャーです。

2 ストライダー飛竜

X68000用　5″2HD版　9,800円(税込)

2名

カプコン



飛ぶ，跳ねる，宙返りなどの多彩なアクション，見応えのあるデカキャラが魅力の作品。

3 プロサッカー68

X68000用

5″2HD版　9,800円(税込)

2名

イマジニア　☎03(3343)8900

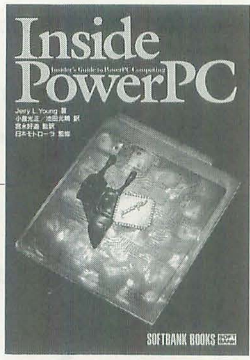


スピーディな試合展開が気持ちいいサッカーゲーム。本格的なリーグ戦も楽しめるぞ。

4 Inside PowerPC

3名

ソフトバンク　☎03(5642)8100



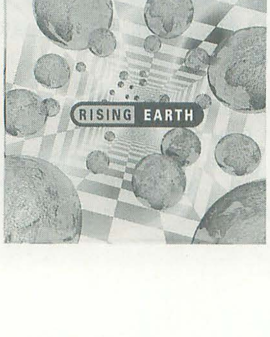
ひょっとして，読んでおくといいことがあるかも(?)，と意味深な言葉とともにお勧めする1冊です。

5 A) MYU-TATION B) RISING EARTH

各8名

MYU-RECORDINGS





「MYU-NET」に集う個性豊かなサウンドが集まった，オムニバス形式のCDです

9月号プレゼント当選者

1同人ソフトセットⅠ(大阪府)嶋村　謙　(神奈川県)河合健一　(神奈川県)中村　正夫　**2**同人ソフトセットⅡ(徳島県)阿部　努　(栃木県)大嶋　靖浩　(大阪府)谷口　浩　**3**ラストバタリオン(埼玉県)伊東　輝之　(茨城県)菅野　崇(長野県)平野　鉄之助　**4**Inside PowerPC(山梨県)小池　克博　(和歌山県)工藤　圭一　(大阪府)宮内　功知　(敬称略)以上の方々が当選しました。商品は順次発送致しますが，入荷状況などにより遅れる場合もあります。

S-OSディスクイメージファイルアーカイバ

SOSAR.X

Itou Masahiko 伊藤 雅彦

S-OSのディスクイメージファイルをアーカイバ感覚で簡単，ラクラク扱えるツールの登場です。従来のSOSCOPY.XとSOSFORMAT.Xの機能も組み込まれた，多機能なものに仕上がっています

X68000ユーザーの皆さん，たまにはS-OSやってますか？ やってるという方に使っていただきたいのがこのツール，SOSAR.Xです。SOSAR.Xは，Human68kのファイルをディスクファイルに転送してS-OSで使えるようにしたり，逆にディスクファイルに記録されたS-OSファイルをHuman68kのファイルにしたりするものです。つまり，Human68kとS-OSとの間でファイルの橋渡しをするわけです。

なお，以前にHuman68kファイルをS-OSファイルにするツールであるSOSCOPYや，ディスクファイルを作成するSOSFORMATが発表されていますが，今回のSOSAR.Xはそれらの機能を含むことになります。

コンパイル方法

ソースはCで書いてあります。XCなら，

CC SOSAR.C /Y

とし，GCCなら，

GCC SOSAR.C -ldos

としてコンパイルします。ただ，LIBC(ver.1.1.31)を使ったところ，RAMディスクに対するファイル操作がうまくいきませんでした(fseek()がコケる)。なるべく標準のライブラリを使用してください。

SOSAR.Xの使い方

このツールの使い方は，基本的にAR.Xのようなアーカイバと同じです。というのも，ディスクファイルにはいくつかのS-OSファイルが記録してあります。ですから，見方を変えれば，ディスクファイルは一種のアーカイブファイルと考えられるわけです。そこで，ディスクファイルというアーカイブファイルにファイルを登録したり，またそこからファイルを抽出したり，といった感覚で使えます。

基本的な書式は，

SOSAR /x sosdisk file1 file2

といった感じになります。オプションスイッチの位置はどこでもかまいません。ファイル名は最初のものがディスクファイル名で，それ以降のものは登録/抽出/削除するファイル名と解釈されます。ディスクファイル名だけは，拡張子を省略すると'.d'が自動的につけ加えられます。

それでは，各機能の使い方を説明しましょう。

●登録・更新(/u)

Human68kのファイルをディスクファイルに登録します。/uを指定しなくても，/x(抽出)，/d(削除)，/l(リスト表示)を指定しなければ，登録処理になります。

ファイルをテキスト転送するか，バイナリ転送するかを選択するために/aまたは/bのスイッチを指定できます。/aならテキスト転送で，改行コード，EOFコードの変換を行います。/bならバイナリ転送，つまり“そのまま”転送します。さらに，テキスト転送で'/a8'などと数値を指定すると，その数値をタブ幅とみなしてタブコードをいくつかのスペースに変換します。転送したファイルをタブコードに対応していないアプリケーションで扱うときには，この機能を使ってください。

転送方式を指定しなかったときには，新規登録の場合はバイナリ転送になり，更新登録の場合は更新前のファイル属性に従います。つまり，たとえばシャーペンでS-OS用のプログラムを書いて，

SOSAR /a sosdisk game.asm

とテキスト転送で新規登録したとします。そして，あとで修正を加えて再び転送するときには，

SOSAR sosdisk game.asm

というように，テキスト転送の指定はいらなくなります。

このほかに使えるオプションとして，/t(ロード先頭アドレスの指定)と/e(実行開始アドレスの指定)があります。オプション省略時には，新規登録ならロード先頭アドレス$3000，実行開始アドレス$1FFAになります。更新の場合は更新前のままです。

また，存在しないディスクファイル名を指定すると，ディスクファイルを新たに作成します。ディスクファイルの作成だけを行いたいときには，

SOSAR newdisk

というように，ディスクファイル名だけを指定すればOKです。

●抽出(/x)

ディスクファイル上のS-OSファイルをHuman68kのファイルに変換します。登録・更新と同様に，/a，/bで転送方式を指定できます。指定しなかった場合には，ファイル属性に従って転送方式を選択します。

●削除(/d)

ディスクファイル上のS-OSファイルを消去します。

●リスト表示(/l)

ディスクファイルに記録されているS-OSファイルのファイル名をリスト表示します。ファイル名がHuman68kでは許されないものだった場合，ファイル名の前に×印がつきます。このようなファイルを抽出したり削除したりすることはできません。

大容量ファイルへの対応

S-OSのファイル管理方法では，64Kバイト以上のファイルのファイルサイズを正確に把握できません。sosarでは，64Kバイト以上のファイルを登録すると，ディレクトリにファイルサイズ0と記録します。そして，このファイルを抽出するときはFATを見て転送サイズを決めますので，ほとんどの場合，ファイル末尾に余計なデータがくっついてしまいます。ただし，テキスト転送ならEOFコードで転送が終わりますので，このようなことはありません。

リスト SOSAR.C

```c
/*
        S-OSディスクイメージファイルアーカイバ "sosar"
                                Program : 伊藤雅彦
*/

#include <stdio.h>
#include <stdlib.h>
#include <io.h>
#include <doslib.h>
#include <string.h>
#include <ctype.h>


void    chkoption(char *);
void    chkdblex(char, char);
int     getdec(char **);
int     gethex(char **, char);
int     hextoi(char);
void    dblopterr(char);
void    mlsopterr(char);
void    update(void);
int     ascupdate(void);
int     binupdate(void);
void    makedf(void);
void    assigndir(void);
unsigned char   getnucl(void);
void    extract(void);
char    ascextract(int);
void    binextract(int);
void    delete(void);
void    list(void);
void    setdfpara(void);
void    chkdfexist(void);
void    prepdf(char);
void    writedf(void);
void    delfat(unsigned char);
int     chksosf(char, char);
int     skipparap(void);
int     sossearch(void);
int     paratofn(char *, char *);
void    sostofn(char *, char *);
void    helpend(void);
void    fileerr(char *);

struct {
        unsigned char   atr,
                        node[13], ext[3],
                        pw,
                        szl, szh,
                        ltl, lth,
                        exl, exh,
                        blkl[6],
                        cl,
                        blk2;
} *sosd;

int     paras;
int     nowpara;
char    **parap;

FILE    *sosfp, *hufp;

char    dfpara[256], dfname[23], hfname[23];

unsigned char   sosfat[0x100], sosdir[0x1000], sosclbuf[0x1000];

char    prmode = 0;
char    trmode = 0;
char    tabw = -1;
int     topadr = -1;
int     exeadr = -1;

int     ascudcol;
char    ascudctrl;

main(int argc, char *argv[])
{
    int     i;

    for (i = 1 ; i < argc ; i++) {
        if (*argv[i] == '/' || *argv[i] == '-')  chkoption(argv[i]);
    }
    if (prmode == 0)  prmode = 'u';
    if (trmode) {
        switch (prmode) {
          case 'x':
            if (trmode == 'a' && tabw != -1) {
                fprintf(stderr, "/aオプションのタブ幅指定は無意味です¥n");
                helpend();
            }
            break;
          case 'd':
          case 'l':
            mlsopterr(trmode);
        }
    }
    if (prmode != 'u') {
        if (topadr != -1)  mlsopterr('t');
        if (exeadr != -1)  mlsopterr('e');
    }

    paras = argc - 1;
    parap = argv;
    nowpara = 0;
    switch (prmode) {
      case 'u':
        update();
        break;
      case 'x':
        extract();
        break;
      case 'd':
        delete();
        break;
      default:
        list();
    }

    exit(EXIT_SUCCESS);
}


void    chkoption(char *optchr)
{
    char    oc;

    while (*++optchr) {
        switch (oc = tolower(*optchr)) {
          case 'u':
          case 'x':
          case 'd':
          case 'l':
            chkdblex(prmode, oc);
            prmode = oc;
            break;
          case 'a':
            tabw = getdec(&optchr);
          case 'b':
            chkdblex(trmode, oc);
            trmode = oc;
            break;
          case 't':
            if (topadr != -1)  dblopterr(oc);
            topadr = gethex(&optchr, oc);
            break;
          case 'e':
            if (exeadr != -1)  dblopterr(oc);
            exeadr = gethex(&optchr, oc);
            break;
          case '/':
          case '-':
            break;
          default:
            fprintf(stderr, "/%cオプションはありません¥n", oc);
            helpend();
        }
    }
}


void   chkdblex(char optchr1, char optchr2)
{
    if (optchr1) {
        if (optchr1 == optchr2) {
            dblopterr(optchr1);
        } else {
            fprintf(stderr, "/%cオプションと/%cオプションは同時に指定できません¥n", optchr1, optchr2);
            helpend();
        }
    }
}


int     getdec(char **optchr)
{
    int     rc;

    if (isdigit(*(++*optchr))) {
        rc = **optchr - '0';
    } else {
        (*optchr)--;
        return(-1);
    }
    while (isdigit(*(++*optchr))) {
        rc = rc * 10 + **optchr - '0';
        if (rc > 99) {
            fprintf(stderr, "/aオプションのタブ幅が大き過ぎます(最大99)¥n");
            helpend();
        }
    }
    (*optchr)--;
    return(rc);
}


int     gethex(char **optchr, char opts)
{
    int     rc;

    if (isxdigit(*(++*optchr))) {
        rc = hextoi(**optchr);
    } else {
        fprintf(stderr, "/%cオプションにアドレスが指定されていません¥n", opts);
        helpend();
    }
    while (isxdigit(*(++*optchr))) {
        rc = rc * 0x10 + hextoi(**optchr);
        if (rc > 0xffff) {
            fprintf(stderr, "/%cオプションの指定アドレスが大き過ぎます(最大ffff)¥n", opts);
            helpend();
        }
    }
    (*optchr)--;
    return(rc);
```

▶(善)さん，僕のソニーのキララバッソ「KV-29ST10」でもリモコンコマンドが設定できました。多謝。
遠藤　勝博(25)宮城県

```
}


int	hextoi(char hx)
{
   if (isdigit(hx))  return(hx - '0');
   return(tolower(hx) - 'a' + 10);
}


void	dblopterr(char optchr)
{
   fprintf(stderr, "/%cオプションが重複しています¥n", optchr);
   helpend();
}


void	mlsopterr(char optchr)
{
   fprintf(stderr, "/%cオプションは無意味です¥n", optchr);
   helpend();
}


void	update()
{
   int	fsize, trsize;
   char	aflg, nflg, wflg;
   unsigned char	clstr, topclstr, preclstr;

   setdfpara();
   if (access(dfpara, 0)) {
      makedf();
   } else {
      prepdf(1);
   }

   wflg = 0;
   while (skipparap() == 0) {
      switch (chksosf(0, 1)) {
         case 0:
            delfat(sosd->cl);
            nflg = 0;
            switch (trmode) {
               case 'a':
                  aflg = 1;
                  break;
               case 'b':
                  aflg = 0;
                  break;
               default:
                  aflg = sosd->atr & 4;
            }
            break;
         case 2:
            assigndir();
            nflg = 1;
            aflg = (trmode == 'a');
            break;
         default:
            continue;
      }
      if (access(*parap, 0)) {
         fprintf(stderr, "¥'%s¥'は見つかりませんでした¥n", hfname);
         continue;
      }

      if ((hufp = fopen(*parap, "rb")) == NULL)  fileerr(hfname);
      clstr = topclstr = getnucl();
      preclstr = 0;
      fsize = 0;
      ascudcol = 0;
      ascudctrl = 0;
      do {
         if ((trsize = aflg ? ascupdate() : binupdate()) == 0)  break;
         if (clstr == 0)  goto capaerr;
         wflg = 1;
         if (fseek(sosfp, clstr * 0x1000, SEEK_SET))  fileerr(dfname);
         if (fwrite(sosclbuf, 1, trsize, sosfp) != trsize)  fileerr(dfname);

         fsize += trsize;
         if (preclstr)  sosfat[preclstr] = clstr;
         sosfat[preclstr = clstr] = ((trsize - 1) >> 8) + 0x80;

         clstr = getnucl();
      } while (trsize == 0x1000);

      if (fsize == 0) {
         if (topclstr == 0)  goto capaerr;
         sosfat[topclstr] = 0x80;
         wflg = 1;
         fsize = 1;
      }
      if (fsize > 0xffff)  fsize = 0;
      sosd->szl = fsize;
      sosd->szh = fsize >> 8;
      sosd->cl = topclstr;
      if (nflg) {
         int	i;
         char	*hfn;

         sosd->atr = aflg ? 4 : 1;
         hfn = hfname - 1;
         for (i = 0 ; i < 13 ; i++) {
            if ((sosd->node[i] = *++hfn) == '.')  break;
         }
         for ( ; i < 13 ; i++) {
            sosd->node[i] = ' ';
         }
         while (*(hfn++) != '.') {}
         for (i = 0 ; i < 3 ; i++) {
            if ((sosd->ext[i] = *(hfn++)) == '¥0')  break;
         }
         for ( ; i < 3 ; i++) {
            sosd->ext[i] = ' ';
         }
         sosd->pw = 0x20;
         i = (topadr == -1) ? 0x3000 : topadr;
         sosd->ltl = i;
         sosd->lth = i >> 8;
         i = (exeadr == -1) ? 0x1ffa : exeadr;
         sosd->exl = i;
         sosd->exh = i >> 8;
         for (i = 0 ; i < 6 ; i++) {
            sosd->blk1[i] = 0;
         }
         sosd->blk2 = 0;
      } else {
         switch (trmode) {
            case 'a':
               sosd->atr = (sosd->atr & 0xf8) + 4;
               break;
            case 'b':
               sosd->atr = (sosd->atr & 0xf8) + 1;
         }
         if (topadr != -1) {
            sosd->ltl = topadr;
            sosd->lth = topadr >> 8;
         }
         if (exeadr != -1) {
            sosd->exl = exeadr;
            sosd->exh = exeadr >> 8;
         }
      }

      if (fclose(hufp))  fileerr(hfname);
   }

   if (wflg)  writedf();
   if (fclose(sosfp))  fileerr(dfname);

   return;

capaerr:
   fprintf(stderr, "¥'%s¥'登録中にディスクファイルの容量が足りなくなりました¥n", hfname);
   if (wflg) {
      delfat(topclstr);
      if (nflg == 0) {
         sosd->atr = 0·
      }
      writedf();
   }
   if (fclose(sosfp))  fileerr(dfname);
   if (fclose(hufp))  fileerr(hfname);
   exit(EXIT_FAILURE);
}


int	ascupdate()
{
   int	trsz, trchr;
   char	*scbp;

   scbp = sosclbuf;
   trsz = 0;
   while (trsz < 0x1000 && ascudctrl != -1) {
      if (ascudctrl) {
         *(scbp++) = ' ';
         trsz++;
         ascudctrl--;
         continue;
      }
      if ((trchr = fgetc(hufp)) == EOF) {
         if (ferror(hufp))  fileerr(hfname);
         *(scbp++) = 0;
         trsz++;
         ascudctrl = -1;
         continue;
      }
      if (iscntrl(trchr)) {
         switch (trchr) {
            case 0x9:
               if (tabw != -1) {
                  if (tabw)  ascudctrl = tabw - ascudcol % tabw;

                  ascudcol = 0;
                  break;
               }
            case 0xd:
               *(scbp++) = trchr;
               ascudcol = 0;
               trsz++;
               break;
            case 0x1a:
               *(scbp++) = 0;
               trsz++;
               ascudctrl = -1;
         }
         continue;
      }
      *(scbp++) = trchr;
      ascudcol++;
      trsz++;
   }
   return(trsz);
}


int	binupdate()
{
```

▶人工知能についての特集を組んでほしい。　小室　肇(21)愛媛県

```
    int   trsz;

    trsz = fread(sosclbuf, 1, 0x1000, hufp);
    if (ferror(hufp))  fileerr(hfname);
    return(trsz);
}


void   makedf()
{
    int   i;

    printf("ディスクファイル¥'%s¥'を新規作成します¥n", dfname);

    i = 0;
    sosfat[i++] = 1;
    sosfat[i++] = 0x8f;
    do {
        sosfat[i++] = 0;
    } while (i < 0x50);
    do {
        sosfat[i++] = 0x8f;
    } while (i < 0x80);
    do {
        sosfat[i++] = 0;
    } while (i < 0x100);

    for (i = 0 ; i < 0x1000 ; i++) {
        sosdir[i] = 0xff;
        sosclbuf[i] = 0;
    }

    if ((sosfp = fopen(dfpara, "w+b")) == NULL)  fileerr(dfname);
    if (fwrite(sosclbuf, 1, 0xe00, sosfp) != 0xe00)  fileerr(dfname);
    if (fwrite(sosfat, 1, 0x100, sosfp) != 0x100)  fileerr(dfname);
    if (fwrite(sosclbuf, 1, 0x100, sosfp) != 0x100)  fileerr(dfname);
    if (fwrite(sosdir, 1, 0x1000, sosfp) != 0x1000)  fileerr(dfname);
    for (i = 0 ; i < 0x4e ; i++) {
        if (fwrite(sosclbuf, 1, 0x1000, sosfp) != 0x1000)  fileerr(dfname);
    }
}


void   assigndir()
{
    int   i;

    sosd = sosdir;
    for (i = 0 ; i < 128 ; i++) {
        switch (sosd->atr) {
            case 0:
            case 0xff:
                return;
        }
        sosd++;
    }
    fprintf(stderr, "ディスクファイルが異常です¥nディレクトリに新規ファイルを登録する余地がありません¥n");
    if (fclose(sosfp))  fileerr(dfname);
    exit(EXIT_FAILURE);
}


unsigned char   getnucl()
{
    unsigned char   cl;

    for (cl = 2 ; cl < 0x50 ; cl++) {
        if (sosfat[cl] == 0)  return(cl);
    }
    return(0);
}


void   extract()
{
    int   fsize, trsize;
    char   aflg, cflg;
    unsigned char   clstr;

    setdfpara();
    chkdfexist();
    prepdf(0);

    while (skipparap() == 0) {
        if (chksosf(1, 0))  continue;
        if (access(*parap, 0) == 0  && access(*parap, 2)) {
            fprintf(stderr, "¥'%s¥'は同名の書き込み禁止ファイルが存在します¥n", hfname);
            continue;
        }

        if ((hufp = fopen(*parap, "wb")) == NULL)  fileerr(hfname);

        switch (trmode) {
            case 'a':
                aflg = 1;
                break;
            case 'b':
                aflg = 0;
                break;
            default:
                aflg = sosd->atr & 4;
        }
        if ((fsize = sosd->szh * 0x100 + sosd->szl) == 0)  fsize = 0x4e000;
        clstr = sosd->cl;
        if (clstr > 0x4f || clstr < 2)  return;

        cflg = 1;
        do {
            if (fseek(sosfp, clstr * 0x1000, SEEK_SET))  fileerr(dfname);
            clstr = sosfat[clstr];

            trsize = 0x1000;
            if (clstr > 0x4f || clstr < 2) {
                cflg = 0;
                if (clstr >= 0x80 && clstr <= 0x8e)  trsize = (clstr - 0x7f) * 0x100;
            }
            if (fsize <= trsize) {
                cflg = 0;
                trsize = fsize;
            }
            fsize -= trsize;

            if (fread(sosclbuf, 1, trsize, sosfp) != trsize)  fileerr(dfname);
            if (aflg) {
                cflg = cflg & ascextract(trsize);
            } else {
                binextract(trsize);
            }
        } while (cflg);


        if (fclose(hufp))  fileerr(hfname);
    }

    if (fclose(sosfp))  fileerr(dfname);
}


char   ascextract(int trsize)
{
    int   scbi;
    char   trchr;

    for (scbi = 0 ; scbi < trsize ; scbi++) {
        if (iscntrl(trchr = sosclbuf[scbi])) {
            switch (trchr) {
                case 0xd:
                    fputc(0xd, hufp);
                    fputc(0xa, hufp);
                    break;
                case 0x9:
                    fputc(0x9, hufp);
                    break;
                case 0:
                    fputc(0x1a, hufp);
                    if (ferror(hufp))  fileerr(hfname);
                    return(0);
            }
        } else {
            fputc(trchr, hufp);
        }
    }
    if (ferror(hufp))  fileerr(hfname);
    return(1);
}


void   binextract(int trsize)
{
    if (fwrite(sosclbuf, 1, trsize, hufp) != trsize)  fileerr(hfname);
}


void   delete()
{
    char   wflg;

    setdfpara();
    chkdfexist();
    prepdf(1);

    wflg = 0;
    while (skipparap() == 0) {
        if (chksosf(1, 1))  continue;
        wflg = 1;
        sosd->atr = 0;
        delfat(sosd->cl);
    }

    if (wflg)  writedf();
    if (fclose(sosfp))  fileerr(dfname);
}


void   list()
{
    int   i;
    char   fn1[23], fn2[23];

    setdfpara();
    if (skipparap() == 0) {
        fprintf(stderr, "無意味なパラメータがあります¥n");
        helpend();
    }
    chkdfexist();
    prepdf(0);

    sosd = sosdir;
    for (i = 0 ; i < 128 ; i++) {
        if (sosd->atr == 0xff)  break;
        if (sosd->atr) {
            sostofn(&(sosd->node), fn1);
```

▶10月号91ページの横山さんのメッセージを見て，「あぁ，アナログの美音を大切にしている人がまだおられるのだ」と安心しました。友人にアナログを馬鹿にする人がいるもので……寂しい。
巣山　智之(23)福岡県

```
            if (paratofn(fn1, fn2) == 0 && strcmp(fn1, fn2) == 0) {
                printf("  ");
            } else {
                printf("×");
            }
            puts(fn1);
        }
        sosd++;
    }
    if (i == 0) printf("登録されているファイルはありません¥n");

    if (fclose(sosfp))  fileerr(dfname);
}


void    setdfpara()
{
    if (skipparap()) {
        fprintf(stderr, "ディスクファイル名が指定されていません¥n");
        helpend();
    }
    strcpy(dfpara, *parap);
    switch (paratofn(dfpara, dfname)) {
        case 1:
            fprintf(stderr, "¥'%s¥'はファイル名として不適切です¥n", dfpara);
            helpend();
        case 2:
            strcat(dfname, "d");
            strcat(dfpara, ".d");
    }
}


void    chkdfexist()
{
    if (access(dfpara, 0)) {
        fprintf(stderr, "¥'%s¥'が見つかりませんでした¥n", dfname);
        exit(EXIT_FAILURE);
    }
}


void    prepdf(char wflg)
{
    char    *tp[2] = {"rb", "r+b"};

    if (wflg && access(dfpara, 2)) {
        fprintf(stderr, "¥'%s¥'は書き込み禁止です¥n", dfname);
        exit(EXIT_FAILURE);
    }

    if ((sosfp = fopen(dfpara, tp[wflg])) == NULL)  fileerr(dfname);
    if (filelength(fileno(sosfp)) != 0x50000) {
        fprintf(stderr, "ディスクファイルのサイズは320KBでなければなりません¥n");
        if (fclose(sosfp))  fileerr(dfname);
        exit(EXIT_FAILURE);
    }

    if (fseek(sosfp, 0xe00, SEEK_SET))  fileerr(dfname);
    if (fread(sosfat, 1, 0x100, sosfp) != 0x100)  fileerr(dfname);
    if (fseek(sosfp, 0x1000, SEEK_SET))  fileerr(dfname);
    if (fread(sosdir, 1, 0x1000, sosfp) != 0x1000)  fileerr(dfname);
}


void    writedf()
{
    if (fseek(sosfp, 0xe00, SEEK_SET))  fileerr(dfname);
    if (fwrite(sosfat, 1, 0x100, sosfp) != 0x100)  fileerr(dfname);
    if (fseek(sosfp, 0x1000, SEEK_SET))  fileerr(dfname);
    if (fwrite(sosdir, 1, 0x1000, sosfp) != 0x1000)  fileerr(dfname);
}


void    delfat(unsigned char clstr)
{
    unsigned char   tmpcl;

    while (clstr < 0x50 && clstr > 1) {
        tmpcl = sosfat[clstr];
        sosfat[clstr] = 0;
        clstr = tmpcl;
    }
}


int     chksosf(char rsw, char psw)
{
    if (paratofn(*parap, hfname) == 1) {
        fprintf(stderr, "¥'%s¥'はファイル名として不適切です¥n", *parap);
        return(1);
    }
    if (sossearch()) {
        if (rsw) {
            fprintf(stderr, "¥'%s¥'は登録されていません¥n", hfname);
            return(1);
        } else {
            return(2);
        }
    }
    if (psw && sosd->atr & 0x40) {
        fprintf(stderr, "¥'%s¥'は書き込み禁止属性で登録されています¥n", hfname);
        return(1);
    }
    return(0);
}


int     skipparap()
{
    do {
        if (++nowpara > paras)  return(1);
        parap++;
    } while (**parap == '/' || **parap == '-');
    return(0);
}


int     sossearch()
{
    int     i;
    char    hufn[18], sosfn[18], *extp;

    *hufn = '¥0';
    strncat(hufn, hfname, ((i = (extp = strchr(hfname, '.')) - hfname) > 13) ? 13 : i);
    strcat(hufn, extp);

    sosd = sosdir;
    for (i = 0 ; i < 128 ; i++) {
        if (sosd->atr == 0xff)  break;
        if (sosd->atr) {
            sostofn(&(sosd->node), sosfn);
            if (strcmp(hufn, sosfn) == 0)  return(0);
        }
        sosd++;
    }
    return(1);
}


int     paratofn(char *para, char *fn)
{
    struct  NAMECKBUF  finf;

    if (NAMECK(para, &finf))  return(1);
    strcpy(fn, finf.name);
    if (strlen(finf.ext) == 0) {
        strcat(fn, ".");
        return(2);
    }
    strcat(fn, finf.ext);
    return(0);
}


void    sostofn(char *sos, char *fn)
{
    int     sps, i;

    i = sps = 0;
    do {
        if ((*(fn++) = *(sos++)) == ' ') {
            sps++;
        } else {
            sps = 0;
        }
    } while (++i < 13);
    fn -= sps;
    *(fn++) = '.';
    sps = 0;
    do {
        if ((*(fn++) = *(sos++)) == ' ') {
            sps++;
        } else {
            sps = 0;
        }
    } while (++i < 16);
    fn -= sps;
    *fn = '¥0';
}


void    helpend()
{
    printf("使用法：sosar ［スイッチ］ ディスクファイル ［ファイル･･･］¥n");
    printf("¥t/u        ファイルの登録・更新（デフォルト）¥n");
    printf("¥t/x        ファイルの抽出¥n");
    printf("¥t/d        ファイルの削除¥n");
    printf("¥t/l        リスト出力¥n");
    printf("¥t/a[タブ幅]  テキスト転送モード。タブ幅指定時はタブコードをスペースに変換¥n");
    printf("¥t          （/u,/x時のみ有効。タブ変換は/u時のみ有効）¥n");
    printf("¥t/b        バイナリ転送モード（/u,/x時のみ有効）¥n");
    printf("¥t/tアドレス  ロード先頭アドレスの指定（/u時のみ有効）¥n");
    printf("¥t/eアドレス  実行開始アドレスの指定（/u時のみ有効）¥n");
    exit(EXIT_FAILURE);
}


void    fileerr(char *fname)
{
    fprintf(stderr, "ファイル操作中にエラーが発生しました¥n");
    perror(fname);
    fcloseall();
    abort();
}
```

▶X68000 PROを「互換機」とかいっていじめるのはやめてください。
川田　洋(20)東京都



第107回

猫とコンピュータ

猫は何度も夢を見る

Takazawa Kyoko
高沢 恭子

コンピュータが日々進歩して，人間にとって使いやすくなってきます。しかし，コンピュータが変わっていくようには簡単に変われないところが人間にはあるようですが。

きびしい暑さがつづいたあと，秋は駆け足でやってきた。ときには肌寒いほどの朝晩の冷気に，あんなに苦しめられたはずの炎天が少しなつかしくなる。

8月の終わりに海外20数カ国でいっせいに発売された「Windows95」の話題は，もうひとつの熱波のように日本にも伝わってきた。11月には国内でも発売される予定だそうだが，あるところでは英語版がすでに売り出されているという話も聞いた。

ペットのような顔で

マルチメディア型のパソコンが人気となって売り上げが伸び，いまはパソコンブームなのだそうだ。

いままでにもブームと感じたことは何回もあった。夫のまわりに若い友人がふえ，ICやハンダの道具が彼の机や整理棚を占拠しはじめたとき。

キーボードのつながった機械が家のなかに並び，書棚から一般書が押し出されてカナと横文字の本がふえていったとき。

大手電機メーカーのパソコンショウがたくさん開かれるようになったとき。パソコン通信が広まりはじめたとき。

MS-DOSが生まれてパソコンはマニアだけのものではないといわれはじめたとき。

それらはすべて，限られた人たちだけが起こしたひとつづきの現象にすぎないことで，いま，ほんとうのパソコンブームがやっとおとずれたのだろうか。

なるほど，多くの人たちが関心をしめしている新しいタイプのマシンは，いままでとはちがう親しみやすさがある。ゲームも絵も音楽も，通信も，マウスでカンタンにできるらしい。1台のなかにたくさんの楽しみが詰まったデンキおもちゃ箱，値段もそれほど高くない。家族みんなで使えるような気がする。

大手スーパーのダイエーが，アメリカのPackard Bell社のマルチメディア型パソコンを売り出すというので，東京の住まい近くのダイエーN店に出かけてみた。

発売日1日前の午後だったが，すでに商品は展示されていた。

Pentium75MHz搭載，8Mバイト，4倍速CD-ROMドライブ。HDD 1Gバイト，FAXモデム内蔵。サウンドボードつき，15インチカラーモニタ装備で188,000円。

各部の仕様を見ると，メインメモリとディスプレイ以外はわが家の負け，しかも価格はわが家の4分の1くらいなのだから，完敗である。

1台買ってみようかと考えた。「Windows95」のプラグアンドプレイ機能に対する準備もあるというし，なにしろ安い。

ただし，いちばん気になるのはディスプレイが15インチであること。マルチメディアタイプのパソコンには，ディスプレイがものをいう。いまわが家でそれより大きいものを使って実感しているだけに，小さすぎはしないかと考える。

だからといってディスプレイだけを交換するわけにはいかない。オリジナルに組み立てられたマシンを，部分的に交換する考えはやめたほうがいいらしい。費用も高くつき，接続がうまくいくかどうかわからないという。

両サイドに細長いスピーカーをつけたディスプレイは，耳のあるペットロボットのようでかわいいけれど，少々頼りないかもしれない。

しかし家庭に置いてみんで使おうというには最適でお買い得，ヒットするような気がする。

紹介のチラシには，「新世代のマルチメディアパソコン」で誰でもすぐ使えると書かれている。新製品が出るときかならず唱えられる，半分ホントで半分ウソのキャッチコピーだ。

「買ったその日からパソコン通信ができます」というのもある。14400bpsのFAXモデムで通信とFAXの送受信ができるそうだ。モデムも内蔵されているばあいはトラブルがほとんどないのだろうか。わが家の新入りのモデムは，メーカーに2度目の入院をしたまま帰ってこない。

ハコものと呼ぶマシン

その翌日，こんどは富士通がマルチメディアタイプのマシンをやはりダイエーで発売すると聞いて，また出かけた。

「あそこにありますよ，値段はこれからつけるところです」。売場の人がいった。

「新聞にあったとおりの値段ですか？」

「そうです。178,000円です」

展示されていたマシンは，インクジェットのカラープリンタがついているが，ディスプレイは14インチだった。そばの段ボール箱に466Cとあり，Pentium搭載ではないようだ。

「なにか，仕様説明書のようなものはありませんか？」と聞くと，「これにはそういうものは一切ありません」と，ハネつけるような答えが返ってきた。ギリギリの廉価で販売するために説明書もないらしい。

現物をじーっとながめて，自宅のマシンルームに置かれたところをイメージしてみた。自分の家のなかで生きるようなパソコンでなければ意味がない。

三重県のパソコンショップY氏によると，いまたくさん売り出されている一体型の新タイプのマシンは，「ハコもの」と呼ばれて

いるそうだ。

セットされていて便利でもあるが困ることも多い。8MバイトくらいではWindowsを使うには無理があるし,136Mバイトまで増設可能などといっても，じっさいの増設にはヤッカイなことも出てきますよという。

任意組立型のマシンも一体型のマシンもそれぞれに特徴がある。家電のような感覚で使うならハコもの，専門的な使いかたをするには必要に応じて組み替えができるマシンがいいのだろう。

パソコンの売り上げが好調といっても，アメリカの家庭での普及率が40%であるのにくらべて，日本では数%だそうだ。みんなが注目するようになったいま，ハコものマシンが期待を裏切らない，誰にでも使えるパソコンになりますように。

猫に学ぶ柔らかさ

新宿のおばあちゃんが，どうもよく眠れなくて困るというのを聞いて，狛江のアニキが,「眠れなくてもどうってことはないと思えばいいんだよ」と教えていた。

ウマイことをいうなあと感心してしまった。眠らなければならないと思うから眠れなくなる。ホンニャアをはじめ猫たちの眠りかたを見ていると，そのとおりだと思う。

猫たちは眠いから寝るのか，寝るから眠くなるのかわからないが，とにかく，すみやかに眠りにつく。あれは人間のように，眠る時間をきめてそれを守ろうなんてしないからだ。自分のからだの調子で，眠くなったら寝る，あるいはヒマだから寝るので，義務を果たしているのではない。

長距離列車に乗る人たちは，みんなよく眠る。ほどよい振動と，エアコンディションの快適さ。目的地に向かっている安心感もあって，しぜんに眠りにさそわれる。

このあいだは空席の多い新幹線で，とうとう和服の女性が，3連の座席のアーム台をみんなハネ上げて，横になって眠ってしまった。

季節の別なくこれほどみんなが眠るのだから，車内での睡眠グッズなどがもっとヒットしてもいいと思うのに，携帯マクラやアイマスクを使って眠るひとはいない。

つまり外出先で眠ることは本来の予定ではない。予定していないから無理なくできる。それに，たぶん眠ってしまうだろうとは思っていても，眠る用意なんかして出かけたくない。準備がないところにウタタ寝のダイゴ味があるのだから。

それにくらべると，就寝時間を守り，寝るための用意を整えて眠るというのは少し堅苦しい。ときにはうまくいかないことがあっても当然だ。

ものごとにきちんとしている人には，眠るべき時間に眠れないのは，とても困ったことになる。そこでなんとかうまく眠ろうとして，なお眠れなくなる。

そういう人たちがまず試してみる方法としては，義務感をなくすこと，眠れないのは眠くないからだと気楽に考えるのがいいらしい。

16進数で眠る法

1日じゅう自由な猫なら，好きな時間に好きなスタイルで寝ることができるが，会社や学校のある人は，やはり限られた夜のあいだに眠らなければならない。

なんといっても夜眠ることが最良の健康のリズムだそうだ。それを知っているマジメな新宿のおばあちゃんは，理想の時間帯に安眠できないことを気にかける。

「例の方法はどうかしらね」と，夫に相談した。

ずっと前に彼が思いついた睡眠法のひとつ，2桁の 16進数を並べていくやりかたである。それも一番終わりから逆にたどっていく方法で，けっこうキキメがある。

00からはじまり，01，02，03，…，0E，0Fとつづき，10，11，12，…，20，21，22，…，FFで終わる16進数。これを最後のFFから逆さに唱えていくのだ。

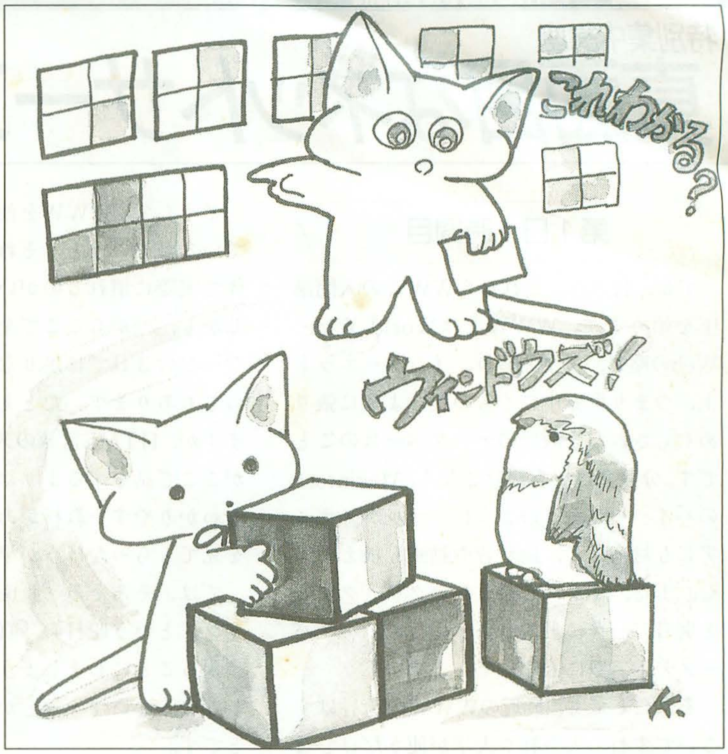


illustration : Kyoko Takazawa

はじめから順序どおり進むのはやさしすぎて，すぐに終わりまでいってしまうが，逆は適度にむずかしく，雑念が除かれるので眠りに入りやすい。

ただし，発案者の夫はこれではカンタンすぎて効果がなくなり，ほかの方法でやることが多いそうだ。

ひとつは，16進数を2進数に置き換える方法。FFは11111111，FEは11111110と，ひとつずつ減らしていく。

つぎは4等分したマス目のなかを塗りつぶして，16進数を表示する方法。1なら右下，2は右上，3は右の上下を合わせた数なので右半分を塗る。左上は4，左下は8で，これらを組み合わせると，0から15までの数字がすべて表せる。

これでA5やBDなどとアタマのなかで並べてシルエットをつくっているうちに，いつか眠ってしまうそうだが，私ならあとの2つの方法ではかえって頭が冴えてきて，紙とエンピツをとりにいくだろう。

数学と理科が大好きなおばあちゃんは集積回路の本も理解できるのだから，この方法は案外ふさわしいかもしれない。

「ヒツジが1匹……」のメルヘンバージョンでは，彼女はほかのことも考えながら，朝まで2万匹くらいヒツジを数えてすごすことだろう。

特別集中講座

真面目なネットサーフィン入門

第1日1時間目

では，皆さん，これからWWWの入門講座を始めます。WWWとはWorld Wide Webの略で，文字どおり，インターネット上，つまり世界中にくもの巣のように張りめぐらされた分散型のデータベースのことです。分散型というだけでなく，Macintoshのハイパーカードのように，マルチメディアにも対応して，絵や音や動画も扱えますし，また，マウスでどこかをクリックすると別のデータに飛ぶというように，ハイパーメディア型にもなっています。

それにしても，最近のWWWの流行はすごいですね。大会社や大学が使うだけでなく，新聞社などのメディアから，街のうどん屋までという時代ですからね。ニュースステーションでもWWWを使ったアンケートをやってますしね。

でも，いちばん面白いのは，単に組織が宣伝のために使うというだけでなく，それらと対等に，一個人が自由に自分をアピールするために使うことができるという点でしょうね。

フランス核実験反対を訴える東大の大学院生数人のWWW上の呼びかけに対して，組織的な力なしに4万人，5万人という人が賛同の署名を行ったというのはきわめて象徴的な，それこそ歴史的な出来事ですね。政治的なことに限らず，社会，文化，経済などあらゆる場面で一個人が主体的な力を出しうる場となる可能性があるのかもしれませんね。

この集中講座では，WWWに関する基礎的な考え方から，最後はWWW上で自由に情報を発信するまでをやってみたいと思います。WWWを使う環境にない人も少なからずいらっしゃると思いますが，今回の講座では，WWWを使えるということを前提として講義を進めますので悪しからず。

いままでは，比較的簡単に家庭からインターネットにアクセスできるようになってきましたし，これからもそういうチャンスはどんどん広がると思います。

どうして，WWWを使える人を対象にしているかというと，それはいうまでもなく，体で実際に慣れるのがいちばんだからです。しかも，一からここで全部お話ししてたら，たったの3日では足りなくなってしまうからでもあります。たとえば，あとで出てきますが，HTML言語の入門に関しては，私がここで話をするよりは，WWW上にとてもわかりやすい教材がありますので，それを見てもらったほうが早いからです。

では，そろそろ「真面目なネットサーフィン」という題目の「真面目な」たる部分に入ることにしましょう。知っておかなければならないマナーとかルールに関することです。

まず，インターネットは世界中の膨大な人がアクセスする場であり，公共性の高い場です。ですから，一般の法律が適用されうる場であるということを強く認識する必要があるのです。

わが国の法律に関連して考えるならば，たとえば，著作権侵害とか，猥褻とか，名誉毀損などについて，意識しなければなりません。実際，インターネットで手に入れた画像を売買して捕まった大学生だっているのです。気をつけてくださいね。

外国でよくても日本で駄目だとか，逆に日本でよくてもほかの国では駄目だとかいうこともあります。要するに，さまざまな価値基準や法律体系をもった国々へのアクセスやそのような国々からのアクセスが自由に行われるインターネットでは，個々人の理性が日常生活以上に問われるということなのです。

では，もうちょっと具体的にまとめておきますので見てください。ここで，ブラウジングというのは，WWWのデータ(ホームページ)を読みにいくことです。ホームページ制作というのは，WWWのデータを自分の計算機の中に書いてそれを公開して，よそからの読み出しを許すことです。

[ブラウジングの際の注意事項]

○技術的な制約

トラフィックを増やすので，画像，音声，プログラムなど，データ量の大きいものはみだりにはもってこない。

[ホームページ制作に関する注意事項]

○技術的な制約

トラフィックを増やすので，画像，音声，プログラムなど，データ量の大きいものはみだりには入れない。

○道徳的な面からの制約

公開を希望しない他人のページやデータ，あるいは，問題のあるページへのリンクはしない。

悪用されたくないデータは入れない。

○本集中講座の主旨からの制約

お金の授受行為につながる表現はしない。

公開を原則とするのでパスワードを入力させて一部の人だけにデータを見せることは望ましくない。

特定の宗教や政治組織，企業利益/不利益に関わる表現はしない。

最後の「本集中講座の主旨からの制約」というのは，この講座の主旨に反することなので，とりあえずやめてくださいということです。一般的なルールでもなんでもないので，誤解のないように。商売をすることはもちろん，仲間うちの連絡にパスワードを入れさせることも，よくあることのようですし。

では，ここでちょっと休憩。

第1日2時間目

それでは，WWWの基礎的な技術を簡単に説明します。まず，図1を見てください。ハイパーテキスト，文章だけでないときは広くハイパーメディアといいますが，これは，複数の文書(データ)がリンクによって結びつけられて構成されています。それぞれの文書はHTML(Hyper Text Markup Language)という言語で記述します。

プログラミング言語というのは基本的にある計算をコンピュータにさせる場合の問題を記述するためにあるものです。しかし，HTMLというのは，MosaicやNetscapeなどのWWWデータを表示させるためのブラウザ(ビューワとか，WWWクライアン

トともいいます)が，そのデータをどのように画面に表示するか，ある部分をマウスでクリックするとどこのデータの表示に移るかとかを規定します。

この図1では，丸く囲ったところに4つのHTML文書があります。そして，HTML文書1の中の「あれこれ」という文字列をクリックするとHTML文書2の表示に移り，また，「xyz」というところをクリックするとHTML文書3の表示に移るわけです。簡単ですね。このような「あれこれ」や「xyz」のことをタグと呼びます。

飛び先として，単に，HTML文書4とするだけでなく，その文書の中の「zzz」と書かれた位置から表示するというように，細かく場所を指定することができます。ですから，同じ文書の中の別の場所を指定することもできます。たとえば，ひとつの文書の中で冒頭に目次があって，そこから各章にリンクするというような使い方はよく見られます。

あと，重要なことは，公開されているWWWデータならたとえ外国の計算機の中のデータでも，まったく同じようにリンクすることができるのです。ですから，マウスをクリックしていると気づかないうちに，別の計算機の中のWWWを見ているなということは頻繁にあります。世界規模のくもの巣はこのようにリンクし合うことによってでき上がっているのです。

では，図2を見てください。今度はWWWデータを見る場合に具体的にはどういう手順を踏んでいるのかを説明します。

皆さんが，MosaicやNetscapeにおいてマウスでクリックして別の文書を読み出すことを考えます。そのとき，HTML文書内には，URLという形式でそのファイルの所在やデータを自分のところにもってくる通信形式が書かれています。たとえば，

http://www.nagoya-u.ac.jp/index.html

と指定すれば，www.nagoya-u.ac.jpで指定されるサイト(計算機)の中のindex.htmlという名前のファイルをhttp方式(html文書の標準的な通信方式)でもってくるということを意味します。

さて，インターネット上を伝わって，www.nagoya-u.ac.jpで表される計算機に要求が伝わるとします。データを公開する計算機上では，要求に応じてデータを送り出すプログラム(WWWサーバ)が動いています。これが，index.htmlという名前のHTML文書を送り返すわけです。要求を送った側にHTML文書が届くと，あとはその文書に書かれているとおりに画面に表示するだけです。

ですから，皆さんが使っている計算機がインターネットに接続されているなら，ブラウザソフトを走らせれば，いわゆるネットサーフィンができて，世界中のデータを読むことができます。さらに，自分でデータを書いてそれを世界中に公開したいなら

図1 HTML文書の概念

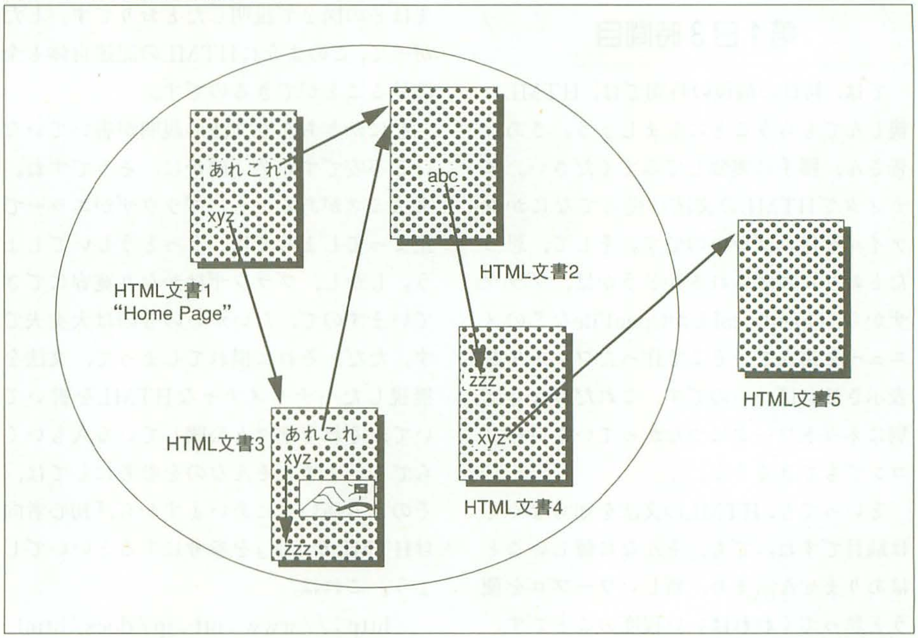
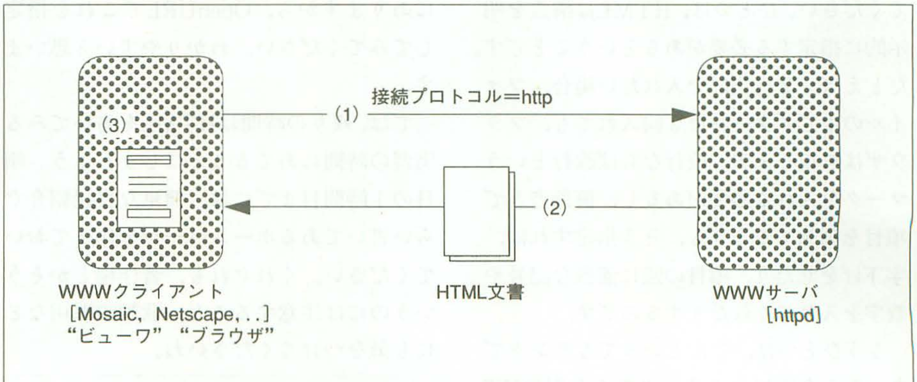


図2 HTML文書転送方式



(1) WWWクライアントはURLによりhtml文書を要求
URL〔Uniform Resource Locator〕
=接続プロトコル://接続先/ディレクトリあるいはファイルのパス
接続プロトコル=http, ftp, gopher, nntp, telnet
(2) 指定されたWWWサーバは指定されたHTML文書などを返送
(3) 返送されたHTML文書を解釈することにより情報を表示

特別集中講座

真面目なネットサーフィン入門

ば，httpdなどのWWWサーバソフトを走らせればいいわけです。

ブラウザだけでなく，サーバソフトもMacintoshなどのパソコンで走るようになってきました。ただ，いつも読めるようにするには，パソコンの電源を1日中つけて，インターネットに接続しておく必要がありますし，別の重いプログラムを走らせると，読み出しに時間がかかってしまうので役不足な気がします。

第1日3時間目

では，初日の最後の時間では，HTMLに親しんでもらうことにしましょう。さあ，皆さん，勝手に実習してみてください。エディタでHTMLの文法に従ってなにかファイルを作ればいいのです。そして，思ったとおりに表示されるかどうかは，ブラウザから，OpenLocalとかOpenFileなどのメニューを選んで，そこで作ったファイルを表示させればいいのです。これだけならば，別にネットワークにつながっていないパソコンでもできますよ。

といっても，HTMLの文法を知らなくては駄目ですね。でも，そんなに難しいことはありません。まあ，新しいワープロを使うと思ってくれればいい程度のことです。

ただ，ワープロとは2点だけ違うと思ってください。ひとつは，HTMLは構造を明示的に指定する必要があるということです。たとえば，2行空白を入れたい場合，ファイルの中でリターンを2回入れても，ブラウザは無視します。改行ならば改行というマークを入れる必要があるし，箇条書きで項目を列挙するときも，そう指定すれば，字下げをしたり，項目の頭に適当な記号や数字を入れてくれたりするのです。

もうひとつは，なんといってもリンクです。ある文字列をクリックすると別のHTML文書などに移る先を指定できることが，ハイパーメディアたるゆえんですからね。これは，URLで指定すればいいのです。

うーむ，これだけでは，なにもわかりませんね。では，近道を教えましょう。それは，人のHTML記述をパクる，ではなかった，参考にすることです。NetscapeでもMosaicでも，View Sourceというコマンドがあります。これは，現在表示しているページのHTML記述を見せてくれるものです。それを参考にして，自分用に修正すれば，でき上がり！　というわけです。

ブラウザで見ているということは，そのHTML文書全体を自分の使っている計算機の中にもってきていることを意味すると，先ほどの図2で説明したとおりです。したがって，このようにHTMLの記述自体も全部見ることができるのです。

なにかきちんと文法の説明が書いていないと不安ですか？　確かに，そうですね。文法ミスがあるごとにブラウザがエラーで止まってしまっては，うっとうしいでしょう。しかし，ブラウザはかなり寛容にできていますので，たいがいのものは大丈夫です。ただ，それに慣れてしまって，文法を無視したハチャメチャなHTMLを書いていて，それを堂々と公開している人もいくらでもいます。そんなのを参考にしては，そのうち痛い目にあいますから，「初心者向けHTMLガイド」を参考にするといいでしょう。これは，

http://www.ntt.jp/docs/html-jman/ncsa-j.html

にありますから，OpenURLでこれを指定してみてください。わかりやすいと思います。

では，残りの時間はHTMLを書いてみる実習の時間にあてることにしましょう。明日の1時間目までには，簡単な自己紹介ぐらい書いてあるホームページを作っておいてください。くれぐれも，著作権とかそういうのには注意すること。歌詞の引用などにも気をつけてくださいね。

第2日1～3時間目

朝イチでくどくど説明しても，皆さん眠そうなので，ここで突然ですが，課題を出しましょう。

＊　＊　＊

課題1

「日本の大学におけるWWWを利用した情報発信の現状」に関係する内容でレポートを作りなさい。レポートはHTMLで記述し，適切なタイトルをつけること。

＊　＊　＊

近くの数人でグループになって，分業でひとつのレポートを作ってください。各メンバーのホームページからひとつのページにリンクを張って，そのページを表紙にしてください。タイトルと名前も表紙のページにつけること。テーマは自由に設定してかまいません。せっかく，HTMLで書くのですから，題材にしている大学のページへのリンクも当然あったほうがいいですね。

あと，大学のページをどうやって見つけるかということですが，そうですね。じゃあ，WWWのページをもっている大学の一覧表が載っている場所を2カ所挙げますから，そこから，興味のある大学をたどって調べてください。

ひとつは，毎日新聞のページの中にあるもので，場所は，

http://www.mainichi.co.jp/server/education/university.html

です。もう，ひとつは，富山大学の中にある一覧表で，場所は，

http://www.toyama-u.ac.jp/hmt/scs/jpnunij.html

です。あとは，テーマと分担を決め，最後に結果を合わせればいいわけです。

昨日はしゃべりすぎて，ノドが痛いのであとはグループごとに自由に作業を進めてください。私は部屋で休んでますので，用があったらメールで知らせてください。アドレスは，

ari@info.human.nagoya-u.ac.jp

です。

そうだ，締め切りをいうの忘れていました。今日の夕方にしましょう。5時きっかりを締め切りとします。丸一日かけて，やれるだけやってみてください。Mosaicを使うのもほとんど初めてという人にいきなり，調査して，レポートをHTMLで書けという

のは，少し無謀という気はしますが，それなりに面白いと思いますので。

(午後5時ごろ自分の部屋で)

5時が締め切りったって，どうやって締め切るんだ？　はて……，順番に見ていったって，簡単に全部は見られないし，5時にまるごとコピーして，どこかにもってきたら，URLが合わなくて，うまく見られなくなってしまうし……。

第3日1時間目

今日は天気もいいことですので，ネットサーフィンの基本的な技術を少し説明することにしましょう。もちろん，次から次へと気分の向くままリンクをたどるというのもいいでしょう。

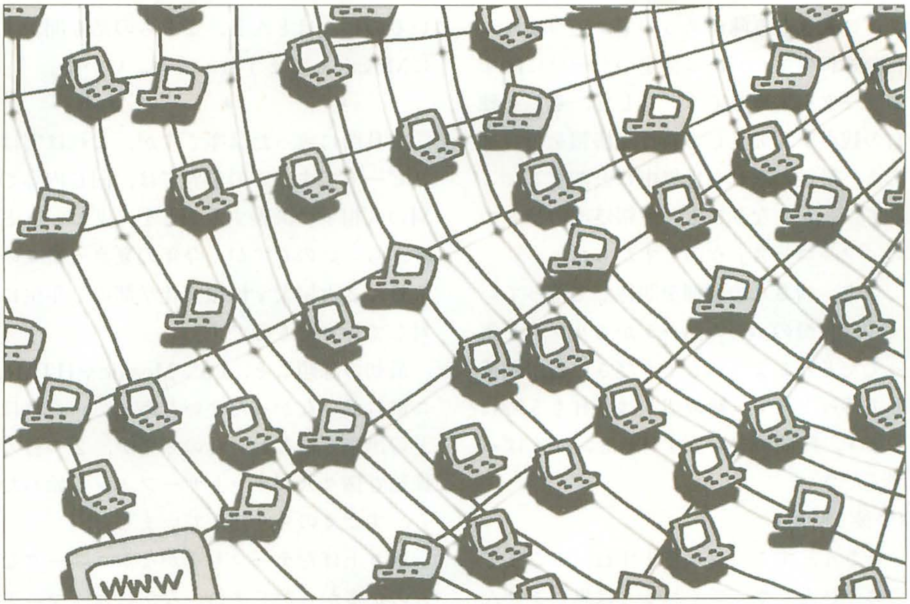
でも，皆さん，考えてもみてください。WWW上のデータなど人が一生かけて朝から晩まで読み続けたとしても，見終わらない分量になります。ですから，基本的な技術をマスターして，ぴしっぴしっと目標のデータにたどり着くことができるようになってほしいですね。

技術といっても，それほど大したことはありませんから，ビビらないように。データを検索するためのページがいろいろと用意されているので，それを利用すればいいのです。

では，ここで，秘蔵のデータを一挙に公開することにしましょう(表)。この夏，私が暇を見つけては集めたものです。少しだけ苦労しました。

たくさんありますが，YahooとWebCrawlerがお勧めです。Open TExtもよさそうですし。じゃあ，ここで練習のために，皆さんにWebCrawlerを使ってもらいましょう。URLを入力してみてください。ここに挙げたリストをページにしてありますので，そのページを出してもらえれば，クリックするだけで行けますよ。

WebCrawlerはサーチエンジン型ですから，検索したいキーワードを入力すると，その単語にマッチしたページが一覧表に出るので，そこでクリックすればすぐにマッ



illustration：Haruhisa Yamada

チしたページのあるWWWサーバにアクセスに行って読み出してくれます。

では，例題として，「野茂」にしましょう。野茂に関係あるページを表示させることができた人は手を挙げてください。

NHK衛星の中継を見るときは必ず打ち込まれるので最近は見ないようにしてるのですが，まあ，それは……，おっ，もう見つけられましたか？　あー，いちばん有名な「トルネードボーイ」のページを見つけましたね。ほかにも，あるでしょ，ね。

WebCrawlerは強力ですよ。“Arita”と試しに入れたら，自分のページがいちはやく登録されていて，ずらっと出てきたのには驚きました。24時間世界中のデータを自動探索してキーワードを登録し続けているのですからね。皆さんのページもそのうち登録されるでしょうから，日本語のページだけでなく英語でも書くといいですよ。

では，しばらくいろいろなページを探してみてください。サーチエンジン型ではなくナビゲータ型もなかなか有効です。あと，(Japan)と書いてあるのは，主に日本のデータが登録されているもので，日本語でキーワードを入れて，探索することもできますから，試してみてください。

第3日2，3時間目

それでは，2つ目の課題を出すことにしましょう。これで終わりです。

＊　　＊　　＊

課題2

次のようなタイトルをつけて，特定のテーマに深く絞り込んでそれに関するWWW上のデータを探し，解説(評論)とともに紹介すること。

＊に興味がある人向けのWWWガイド

＊について知りたい人向けWWW案内

＊に関するWWW情報について

いわゆる「リンク集＋α(こっちが重要)」をHTMLで書くこと。あまり知られていないようなデータを自分で発掘し，なるべく海外のデータを中心にすること。

＊　　＊　　＊

2日目の課題とは違って，完全に趣味の世界のことで結構です。自分はこれに関しては通だとか，ファンだとか，いろいろあると思いますけれど，それに関して，前の時間で示したようなデータベースを使って効率的にデータを探してみてください。

探すだけでは駄目ですよ。そのデータをよく吟味して，自分の言葉で評論や解説を加えてください。新たな情報を付加すると

特別集中講座

真面目なネットサーフィン入門

ころに大きな意味があるのです。つまり，自分で世界中からある特定のテーマに関するデータを集めてくる，そして，それを独自の観点から加工して新たな情報を付け加える，最後にそれを世界中に公開する。これで，まがりなりにも，情報処理のひとつの「フルコース」なのですからね。

さあ，皆さんの奮闘を期待して待っています。自分のホームページからリンクを張ってたどれるようにしてくださいね。期待して待ってます。締め切りは今日も5時にします。時間がないですけれど，がんばってください。

(午後5時頃)

皆さん，すごく凝ってますね。ここまでのめり込んでやってくれるとは思いませんでした。ご苦労さまでした，と私が話しているのに，ほとんど，こちらの話も聞かずにMosaicしてますね。まあ，いいや。

＊　　＊　　＊

3日間に渡った講座ですが，これは実はリピートです。この業界では，同じ内容で別の人相手に講義することを，リピートと呼ぶらしいのですが，今年の夏真っ盛りの8月に名古屋大学情報文化学部の2年生に対して行ったものです。

最初の課題こそ，まだ，MosaicやHTMLに慣れていなかったせいもあって，それほどの出来ではなかったのですが，2つ目の課題で個々人がネットサーフィンし始めたら，すごくのめり込んでいました。

取り上げたテーマもなかなかユニークなものが多かったですね。なかには，マニアックというかカルト的なものもあり，漫画関係で「銃夢Home Page in Japan」などというのを作った学生もいました。ほとんどの学生が，WWWやMosaic，HTMLを触ったこともなかったのに3日間でここまでできたというのは，なかなか面白かったのでしょう。見たい方は，

http://133.6.140.166/sis/lecture.html

にありますので，ご自由にどうぞ(ドメイン名の手続きに手間取っているせいもあってIPアドレスむきだしですみません)。

学生の作品だけでなく，紹介したサーチエンジンなども登録してあります。初日に示した注意事項をくれぐれも忘れないで，おもいっきり自由でぶっとんだページを皆さんも書いてみてください。では，3日間ご苦労さまでした。

表　データを検索するためのページ

ナビゲータ型

大項目から小項目へと分類がなされており，最終的に目的のページにたどり着けます

Yahoo：　http://akebono.stanford.edu/yahoo/
The Whole Internet Catalog：　http://www.gnn.com/wic/
EINet Galaxy：　http://lmc.einet.net/
The World-Wide Web Virtual Library：　http://www.w3.org/hypertext/DataSources/bySubject/Overview.html
WWW Virtual Library (Japan)：　http://www-student.ulis.ac.jp/html/virtual-library/
CSJ Yellow Page (Japan)：　http://www.csj.co.jp/J/Yellowpage-j/yp.html
CUI W3 Catalog：　http://cuiwww.unige.ch/cgi-bin/w3catalog
Starting Point：　http://www.stpt.com/WWW Power Index
http://www.webcom.com/power/index.html

サーチエンジン型

探索したい事項をキーワードとして入力することにより目的のページにたどり着けます

WebCrawler Searching：　http://webcrawler.com/
Open Text：　http://www.opentext.com:8080/
Yahoo Search：　http://www.yahoo.com/search.html
Lycos Search Form：　http://lycos.cs.cmu.edu/lycos-form.html
WWWW-WORLD WIDE WEB WORM：　http://www.cs.colorado.edu/home/mcbryan/WWWW.html
Waseda-search(Japan)：　http://www.info.waseda.ac.jp/search.html
WAVE search (Japan)：　http://www1.sony.co.jp/InfoPlaza/WAVESearch/
ALIWEB Search Form：　http://web.nexor.co.uk/public/aliweb/search/doc/form.html
NIKOS Web Search：　http://www.rns.com/cgi-bin/nikos
JumpStation II Document Search　http://js.stir.ac.uk/jsbin/jsii?jsii_doc_search
Search at EINet Galaxy：　http://galaxy.einet.net/www/www.html
Jughead：　gopher://liberty.uc.wlu.edu:3002/7
ALIWEB Search Form：　http://web.nexor.co.uk/public/aliweb/search/doc/form.html
veronica：　gopher://veronica.scs.unr.edu/11/veronica
directory-of-servers index：　http://www.digital.com:8081/cnidr.org:210/directory-of-servers
GNA Meta Library Search：　http://uu-gna.mit.edu:8001/cgi-bin/meta
ArchiePlex：　http://cuiwww.unige.ch/./archieplexform.html

ガイド型

特定の目的に絞ったページを推薦してくれます

Cool Site of the Day：　http://www.infi.net/cool.html
First Pointers for a Women's Guide：　http://mevard.www.media.mit.edu/people/mevard/women.html
SCOTT'S INTERNET HOT LIST：　http://www.contrib.andrew.cmu.edu/usr/mk42/rapier/scotlist.html
What's New in Japan：　http://www.ntt.jp:80/WHATSNEW/index-j.html

地形型

地図や地域名を選択することにより目的のページにたどり着けます

Virtual Tourist World Map：　http://wings.buffalo.edu/world/
World-Wide Web Servers: Summary：　http://www.w3.org/hypertext/DataSources/WWW/Servers.html
Clickable W3 Map(Japan)：　http://www.ntt.jp/japan/map/index.html

メタデータベース

サーチエンジン型のデータベースが集まっています

W3 Search Engines：　http://cuiwww.unige.ch/meta-index.html
W3 Search Engines(東大)：　http://web.yl.is.s.u-tokyo.ac.jp/net/meta-index.html
Internet Resources Meta-Index：　http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/MetaIndes.html

バックナンバー案内

BACK ISSUES

ここには1994年11月号から1995年10月号までをご紹介しました。現在1994年8～12月号，1995年4～6，9，10月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は128ページを参照してください。

1994



11月号
特集　STEP UP BASIC
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
TeX入門講座/DōGA CGアニメーション講座
システムX探偵事務所/ローテクエ作/善バビ
●新製品紹介　BJC-400J/X680x0 Develop. & libcII
Free Software Selection Vol.2
LIVE in '94　ダーク・スペース/ENDLESS RAIN/レナのテーマ
THE SOFTOUCH　スーパーストII/餓狼伝説SPECIAL
全機種共通システム　B-GALETS2



12月号
特別企画　XL/Imageお試し版＋α
連載　響子in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテクエ作/TeX入門講座
●特別付録　XL/Imageお試し版＋α(5″2HD)
●新製品紹介　H.A.R.P/XDTP SX-68K
LIVE in '94　幻想即興曲/きまぐれ　オレンジ☆ロード　他
THE SOFTOUCH　魔法大作戦/スーパーストII
全機種共通システム　シューティングゲーム作成講座(4)

1995



1月号(品切れ)
特集　割り切って使うCD-ROM
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
ファイル共有の実験と実践/DōGA CGアニメーション講座
システムX探偵事務所/ローテクエ作/TeX入門講座
●CD-ROMドライブ紹介　CS-CD301X/CDS-E/SCD-200
●新製品紹介　X68000XVI用アクセラレータXellent30
LIVE in '95　ぷよぷよ/ジムノペディNO.1/PRIME
THE SOFTOUCH　バックランド/上海　万里の長城/魔法大作戦
餓狼伝説SP 特別編/スーパーストII 特別編



2月号(品切れ)
特集　MicroProcessingUnit
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
SX-BASIC公開デバッグ/DōGA CGアニメーション講座
システムX探偵事務所/SX-WINDOWによるDTP
●特別企画　最新ゲーム機を見る
●新製品紹介　Datacalc SX-68K/シャーペンワープロパック
●1994年度GAME OF THE YEARノミネート作品発表
LIVE in '95　サムライスピリッツ/AFTER SCHOOL/白鳥の湖
THE SOFTOUCH　スーパーストII　特別編



3月号(品切れ)
特集　SoundEffects
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
システムX探偵事務所/ファイル共有の実験と実践
ピコピコエンジン活用講座/SX-WINDOWによるDTP
●SX-WINDOW用ユーティリティ　どっち.X
LIVE in '95　魔法のプリンセスミンキーモモ/別れの曲
ファイナルファンタジーII/宇宙戦艦ヤマト完結編
THE SOFTOUCH　ディグダグ/ディグダグII/VIEW POINT
全機種共通システム　S-OSシステムコールライブラリ



4月号
特集　Let's Play Wonderful GAME
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D
システムX探偵事務所/ファイル共有の実験と実践
DōGA CGアニメーション講座/ローテクエ作
●1994年度GAME OF THE YEAR発表
●新製品紹介　TS-6BSImkII/MJ-5000C/MATIER ver.2.1
LIVE in '95　天聖龍/ファイナルファンタジーVI/
ANOTHER DAY/ハートオブザマッドネス
全機種共通システム　S-OSねちねち入門(1)



5月号
特集　Realize Graphic
連載　響子 in CGわ～るど/ショートプロぱーてぃ
ローテクエ作実験室/SX-BASIC公開デバッグ
システムX探偵事務所/ANOTHER CG WORLD
●特別付録　Oh!電脳倶楽部
●新製品紹介　フォント＆ロゴデザインツール
LIVE in '95　ドラゴンセイバー/ミッドナイトレジスタンス　他
THE SOFTOUCH　ボンバーマン　ぱにっくボンバー
全機種共通システム　S-OSねちねち入門(2)



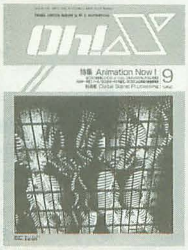
6月号
特集　Open the SX-WINDOW
連載　響子 in CGわ～るど/ハードコア3Dエクスタシー
DōGA CGアニメーション講座/ローテクエ作実験室
システムX探偵事務所/ショートプロぱーてぃ
●特別企画　X68000周辺機器パワーアップ計画
●新製品紹介　Xellent30s/学研統合電子辞書 for SX-Window
●第6回アンケート分析大会
LIVE in '95　クリティカルポイント/THE SUMMER OF '68 他
全機種共通システム　S-OSねちねち入門(3)/BLOCK DOWN



7月号(品切れ)
特集　Optimizing Method
連載　響子 in CGわ～るど/ハードコア3D/ファイル共有
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ANOTHER CG WORLD
●THE USER'S WORKS SPECIAL
●新製品紹介　PDドライブLF-1000
THE SOFTOUCH　バラデューク
LIVE in '95　クロノ・トリガー/SUPER MARIO BGM集 他
全機種共通システム　FE ver.1.0



8月号(品切れ)
特別企画　暑中見舞いPRO-68K
連載　響子 in CGわ～るど/(善)のゲームミュージック
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ANOTHER CG WORLD
●特別付録　暑中見舞いPRO-68K(5″2HD)
●新製品紹介　SCSI2ボードMach-2/DSPボードAWESOME-X
CD-ROMドライブCDG-TX4
LIVE in '95　淡紅色の夢/Tomorrow never knows 他
全機種共通システム　IF ONLY



9月号
特集　Animation Now!
連載　響子 in CGわ～るど/ハードコア3D/DSP
DōGA CGアニメーション講座/ショートプロぱーてぃ
システムX探偵事務所/ローテクエ作実験室
●音声波形表示プログラム　OCR.X
LIVE in '95　ファイナルファンタジーV/SAY ANYTHING
ときめきメモリアル/ドラゴンスレイヤーVI 他
全機種共通システム　FE ver.1.0ラインプリントルーチン詳細
MISSILE SYSTEM



10月号
特集　Now Printing
連載　響子 in CGわ～るど/ショートプロぱーてぃ
Digital Signal Processing/Lisp一夜漬け
こちらシステムX探偵事務所
●新製品紹介　SX-WINDOW ver.3.1開発キット
マルチシンクモニタPC-TM151
LIVE in '95　バイオミラクルぼくってウパ！/闇の血族
ツインビーヤッホー！/TIME STREAM
全機種共通システム　パズルゲームCUBE

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

マルチメディアテレビ
21MM3S
東芝

21MM3S



東芝はマルチメディアテレビ「21MM3S」を発売した。

同機はテレビが映るマルチスキャンディスプレイである。画面サイズは21型で，マスクピッチは0.55mm。水平周波数は15～40kHz，垂直周波数は50～90Hzに対応している。RGB入力端子はD-sub15ピン3列でX680x0への接続は変換コネクタが必要で，映像と音声に対応している。ほかの入出力端子はSビデオ入力，ビデオ入力，オーディオ出力を装備している。音声出力は実用最大出力5Wのスピーカーを2個搭載し，バズーカシステムを採用している。付属品はRGBケーブルやリモコンなどがある。また，パソコンやテレビ，ビデオなどの切り替えや，パソコン映像の表示サイズや位置の微調整はリモコンで行える。

価格は150,000円(税別)。

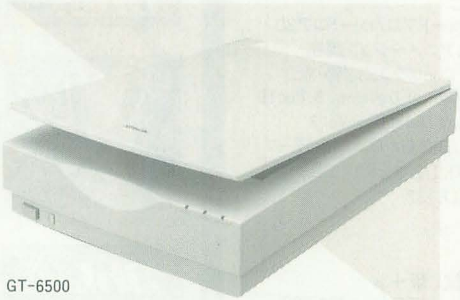
〈問い合わせ先〉

㈱東芝　☎03(5484)5423

カラーイメージスキャナ
GT-5000WINP/WINS/ART
セイコーエプソン

セイコーエプソンはカラーイメージスキャナ「GT-5000WINP」「GT-5000WINS」「GT-5000ART」を発売した。

同機は「GT-6500」シリーズの基本機能を引き継ぎながら大きさ，重さともに3分の2以上のコンパクト化を実現した。大きさが297mm(幅)×443mm(奥行)×87mm(高さ)で，重さが約5kg。解像度は300dpiで，読み取り最大有効領域が216mm×297mmで原稿サイズはA4まで対応。原稿は50～200%のズームが可能。色分解は光源による3原色(RGB)切り替えで，読み取り階調は各色8ビット(最大256階調)，明度調整は7レベルとなっている。



GT-6500

「GT-5000WINP」はインタフェイスが双方向パラレル，「GT-5000WINS」はSCSI，「GT-5000ART」はMacintosh用でインタフェイスがSCSI。

価格は3機種とも59,800円(税別)。

〈問い合わせ先〉

エプソンインフォメーションセンター

☎0424(99)7133，06(399)1115

マッハジェットプリンタ
MJ-3000CU/1100
セイコーエプソン

MJ-3000CU



セイコーエプソンはマッハジェットプリンタ「MJ-3000CU」「MJ-1100」の2機種を発売した。

「MJ-3000CU」はハガキサイズからA2用紙まで対応し，標準装備のオートシートフィーダで給紙ができる。トラクタユニットも標準で装備しているので連続紙への印刷も可能。また，アプリケーションが対応していればA2幅で長さ最大5mまでの用紙に対応できる。

標準ではモノクロ印刷のみの対応で，解像度は720dpi。別売りのカラーアップグレードキット(20,000円)を装着することで普通紙に360dpi，スーパーファイン専用紙に760dpiでのカラー印刷が可能になる。

モノクロインクカートリッジは1回の交換でA4用紙約1700枚(ANK1500字/枚)を印刷できる。印字速度は標準モードで漢字全角が133字/秒，高速モードで266文字/秒。

インタフェイスはパラレルと拡張スロットを1基ずつ装備している。

大きさは666mm(幅)×504mm(奥行)×202mm(高さ)，重さは約11.5kg。

「MJ-1100」はハガキサイズからA3用紙に対応し，内蔵のフロントオートシートフィーダでA4用紙なら連続100枚の給紙が可能。トラクタユニットも標準装備しているので連続紙にも対応。印刷はモノクロのみで解像度は360dpi。印字速度は標準モードで110字/秒(漢字全角)，高速モードで167字/秒を実現している。

大きさは665.6mm(幅)×340mm(奥行)×164mm(高さ)，重さが約9kg。

価格は「MJ-3000CU」が128,000円で「MJ-1100」が79,800円(それぞれ税別)。

〈問い合わせ先〉

エプソンインフォメーションセンター

☎0424(99)7133，06(399)1115

液晶タッチメモ
PA-B2
シャープ

シャープは液晶タッチメモ「PA-B2」を発売した。

同機はデータの入力をペンで行う「PA-B1」の後継機である。各機能はそれぞれに独立したモードキーが採用され，ワンタッチで変更が可能。主な機能はスケジューラー，アクションリスト，電話帳，メモ，電卓などが用意されている。また，スケジューラーを助ける機能として，予定の期日までの日数計算や，予定の準備に必要な日数から



PA-B2

期日を算出する日数計算機能がある。入力方法は従来機の50音入力に加え，ローマ字入力もできるようになった。また，入力しやすいようにペンも従来機に比べ9.5mm長く，0.5mm太くなっている。記憶容量は32Kバイト(ユーザーエリアは約24Kバイト)。

表示部はFSTN液晶(横96ドット×縦64ドット)を採用し，漢字8文字×5桁の表示が可能。大きさは122mm(幅)×82mm(奥行)×13mm(厚さ)で，重さが114g(電池含む)。

価格は11,000円(税別)。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221,03(5261)7271

ペン入力電子手帳

RX-300/500/550/800

カシオ計算機



RX-800

カシオ計算機はペン入力電子手帳「RX-300」「RX-500」「RX-550」「RX-800」を発売する。

「RX-300」は表示部に反射型STNカラー液晶ディスプレイを搭載し，オレンジ，緑，青の3色が表示できる。データの入力はペンで行い，日本語は50音入力やローマ字入力ができる。手書き文字認識機能はないが，画面にペンで書いたメモを記憶できる手書きメモ機能がある。画面は必要に応じて便箋の罫線や方眼紙などを画面上に表示できるので，バランスのとれた手書きが可能。

また，メモ機能には時刻表やゴルフスコアなど5種類の定型メモがある。電話帳，TODO(アクションリスト)，スケジューラ，約82,000語の漢字辞書，パスワードが設定できるシークレット機能，オリジナルの文字や記号を登録できる外字登録などの機能が用意されている。ほかにも，通信機能として同シリーズ間でのデータ送受信やPC-98シリーズとの文字データの送受信が可能。記憶容量は64Kバイト。

「RX-500」は「RX-300」の機能に加え，英和辞典(約40,000語)と和英辞典(約31,000語)があり，「RX-550」は国語辞典(約50,000語)がついている。「RX-800」は「RX-300」の機能に，国語辞典，英和/和英辞典がつき，記憶容量を128Kバイトにしたものである。

価格は「RX-300」が16,500円，「RX-500」が22,000円，「RX-550」が22,000円，「RX-800」が26,000円(それぞれ税別)。

〈問い合わせ先〉

カシオ計算機㈱　☎03(3347)4811

光変調方式採用

オーバーライト光磁気ディスク

日立マクセル

オーバーライト光磁気ディスク



日立マクセルは光変調方式により，オーバーライトを可能にした3.5インチ光磁気ディスク(230Mバイトタイプ)のOEM供給を開始した。

従来の磁界変調方式ではデータ記録を行う際に事前にデータを消去しておく必要があった。光変調オーバーライト方式は磁性多層膜を用い，ディスクへのレーザー照射光強度を2段階に変調することで，すでに記録されたデータの上に直接新しいデータの重ね書きを行う方法である。したがって，消去プロセスが不要となるため従来の光磁気ディスクに比べ原理的に約2倍の転送速度を実現できる。市販は来春の予定。

〈問い合わせ先〉

日立マクセル㈱　☎03(5467)9334

プリンタ専用普通紙

LC-301

キヤノン



LC-301

キヤノンはフルカラーバブルジェットプリンタ専用普通紙「LC-301」を発売した。

同品は同社のプリンタBJシリーズなどのインクジェットプリンタに対応し，水に濡れても滲みがなく，インクの乾きが早いという特長をもっている。また，用紙の両面に印刷ができ，専用コート紙に匹敵する高画質を実現した。A4サイズ1枚当たりの価格が8円とコート紙に比べ約2分の1(同社比)。

価格はA4サイズ1冊(100枚)が800円(税別)。

〈問い合わせ先〉

キヤノン㈱　☎043(211)9530

INFORMATION

文部省認定

CG検定

CG-ARTS協会

財団法人画像情報教育振興協会(略称：CG-ARTS協会)は画像情報技能検定CG部門(略称：CG検定)の1級と3級を以下の要項で実施する。

・出願受け付け期間：

9月1日～10月20日(消印有効)

・実施日：11月26日

・試験内容

○1級：3次試験まであり，筆記，自作品の提出，実技によって判定し，コンピュータグラフィックの専門的な知識を求める

○3級：マークシートのみで，コンピュータグラフィックの初歩的な事柄について理解を求める

〈問い合わせ先〉

CG-ARTS協会　☎03(3535)3501

e-mail：cg-exam@cgarts.or.jp

FILES Oh!X

このインデックスは，タイトル，注記——著者名，誌名，月号，ページで構成されています。秋晴れの空の下でスポーツ，レッツゴー(死語?)。

一般

▶特集1　もう一体型パソコン選びに迷わない
Windowsマシン7機種とMacintosh 4機種の一体型マシンをいくつかの項目でチェックする。——荻窪圭＋中井紀之，ASAHIパソコン，9・15号，16-28pp.

▶特集2　潜入！　オジさんたちの夏パソコン特訓教室
オジさん限定のパソコン講座を編集部員が実際に体験してレポートする。——編集部，ASAHIパソコン，9・15号，30-33pp.

▶98ユーザーのためのマッキントッシュ教室　20
前回に引き続き，PC-98とMacintoshのバスやインタフェイスなど規格の違いについて解説する。——荻窪圭，ASAHIパソコン，9・15号，94-97pp.

▶Global Interface Column
1995年2月にアメリカでハッカーが逮捕された事件のレポート。——笠原利香，ASAHIパソコン，9・15号，118-121pp.

▶インターネットの心
千葉大学の土屋教授を迎え大学側から見たインターネットの状況のレポートや，ホームページ紹介など。——編集部，LOGIN，18号，202-203pp.

▶特集　ファンシープリチーダイナマイツ Windows95デビュー!!
Windows95の3.1からの変更点や現在開発されているゲームソフト，Windows95で動作する3.1対応のゲームやアプリケーションの紹介など。——編集部，LOGIN，18号，211-225pp.

▶くねくね科学探検隊　第26回
数回に分けて無意識について考える。今回は昔の人が無意識についてどう考えていたか紹介する。——鹿野司，LOGIN，18号，272-275pp.

▶特集　パソコン新時代のパーフェクトガイド
Windows95の登場によって変わる，ハードやソフトを予測。——編集部，コンプティーク，10月号，19-29pp.

▶こだわりゲーム年代記
今回はソフトハウス「光栄」からパソコンゲームの歴史的背景や魅力を探る。——与志田拓実，コンプティーク，10月号，92-93pp.

▶NEWS COLLECTORS
任天堂のウルトラ64の初心会での展示を予測したり，通信対戦を可能にする「X-BAND」の紹介など業界の最新情報。——編集部，電撃王，10月号，24-29pp.

▶特集　美少女ソフトカタログ
1995年の4月〜9月までに発売された美少女ソフトを紹介する。——編集部，電撃王，10月号，35-56pp.

▶特集　マルチメディア徹底活用法
デジタルカメラやカラープリンタ，Zipドライブなど各種周辺機器を紹介する。——編集部，マイコンBASIC Magazine，10月号，27-33pp.

▶夢を実現するならこの学校！
コンピュータが学べる学校をゲームクリエイタースクールを中心に紹介する。——編集部，マイコンBASIC Magazine，10月号，48-52pp.

▶Arcade Game Graffiti　第19回
1983年に発売されたアーケードゲームの紹介。今回は「リブルラブル」「バイオアタック」など。——編集部，マイコンBASIC Magazine，10月号，158-159pp.

▶サターン vs プレイステーション
これから発売されるソフト紹介を中心に，どちらが買い得か考える。——編集部，マイコンBASIC Magazine，10月号，168-173pp.

▶特集1　インターネット生活完全ガイド
ネットサーフィンの実際やアクセスガイド，インターネット本の紹介など。——編集部，ASAHIパソコン，10・1号，18-30pp.

▶小田嶋隆のout of byte　第1回
アキバの謎を解明していく連載。第1回はマハーポーシャについて。——小田嶋隆，ASAHIパソコン，10・1号，126-127pp.

▶98ユーザーのためのマッキントッシュ教室　21
Macintosh用の一太郎を使ってワープロについて考える。——荻窪圭，ASAHIパソコン，10・1号，128-129pp.

▶特集2　こうすれば話題のビデオCDが見られます
ビデオCDの簡単な説明とDOS/VとMacintoshのMPEGカードのインストールガイド。——編集部，ASAHIパソコン，10・1号，134-141pp.

▶HardwareForumX
Pentiumの簡単な解説とPC-9821Xa7の改造手順を公開。——編集部，LOGIN，19号，212-215pp.

▶特集　パソコンゲーム黙示録
パソコンゲームの過去を分析し，Windows95を中心に今後のパソコンゲームを予測する。——編集部，LOGIN，19号，219-233pp.

▶ゲーム考現学　第1回
ゲームについての疑問に答える連載。今回は読者アンケートからいま遊びたいRPGを考える。——編集部，LOGIN，19号，238-243pp.

▶くねくね科学探検隊　第27回
無意識のイシキについて考える2回目の今回は昔の人が考えた心理療法について解説する。——鹿野司，LOGIN，19号，284-287pp.

▶パソコン拡張術［基本編］
ボードの交換やハードディスクを増設するなどパソコンを拡張するときの約束ごとを紹介する。——田嶋孝行，I/O，10月号，31-34pp.

▶特集　目的別購入ガイド
ビデオ編集，海外ゲームで遊ぶ，卒業論文作成など目的別の購入ガイド。——森羅万象ほか，I/O，10月号，41-56pp.

▶DTM入門
MIDIの解説や音源ガイド，ソフト紹介など。——依光英世，I/O，10月号，57-69pp.

▶「SCSI-3」次世代規格大研究
SCSI-3で拡張される機能や技術について解説する。——Processor's Professor，I/O，10月号，107-111pp.

▶MultiMedia Watching　22
アメリカのテレビネットワーク買収，CGソフトの動向などマルチメディア周辺の話題を紹介する。——奥野雅之，I/O，10月号，129-131pp.

▶DeskTopMusic入門　7
今回は音楽を入力するための方法をガイドする。——あまだたかし，I/O，10月号，132-135pp.

▶特集I　全解！　かっとびビデオカード
これからのビデオカードを予測したり，各社のビデオカードやユーティリティを徹底比較。——編集部，ASCII，10月号，259-282pp.

▶Abyss of Technology　第2回
「パノラマムービー」をテーマに，そのテクノロジーを解説する。——編集部，ASCII，10月号，283-288pp.

▶特集II　パソコンがグングンつながる！
パソコンをつなぐをテーマに，ネットワークやPeer to Peer，赤外線通信などについて解説する。——編集部，ASCII，10月号，295-310pp.

▶Radical Creative Studio
3DCGソフトの比較レビューやレンダウェアの紹介など。——編集部，ASCII，10月号，377-384pp.

▶Digital Beat Zoo
今回はカシオ計算機のGM音源を紹介する。——編集部，ASCII，10月号，385-388pp.

▶脳型コンピュータを作る　第3回
今回は生理学研究から見た脳について解説する。——松本元，ASCII，10月号，389-396pp.

▶パソコンの周辺機器の作り方全部教えます！
TFTカラー液晶パネルの製作工程やPCカード製作工場を紹介する。——編集部，ASCII，10月号，405-409pp.

X1/turbo/Z

X1turboシリーズ

▶移植版　NEVER ENDING SEA
雲の上を跳ねていくアクションゲーム。——HELL，マイコンBASIC Magazine，10月号，117-118pp.

X1シリーズ

▶MSX2版　ドラゴンスレイヤー英雄伝説〜ユーザーディスク作成BGM〜
FM音源ボード＋NEW FM音源ドライバ用の音楽プログラム。——ふっさん，マイコンBASIC Magazine，10月号，126-127pp.

X68000

▶電撃新作予定表
機種別の予定表。X68000用は「EXCITINGみるく」「プリンセスメーカー」が発売予定。——編集部，電撃王，10月号，187-188pp.

▶カエルの冒険
カニを避けながらゴールに向かう。——高橋秀之，マイコンBASIC Magazine，10月号，119-121pp.

▶FINAL FANTASY VI〜ティナのテーマ〜
NAGDRV＋GS音源用の音楽プログラム。——渋川大吾，マイコンBASIC Magazine，10月号，126-129pp.

▶SUPER SOFT Hot Information
X68000用は「亜美ちゃんの写真集」など。——編集部，マイコンBASIC Magazine，10月号，とじ込み付録11p.

▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたプログラムを紹介する。X68000用はQuickTime再生プログラム「qt_play.x」。——編集部，ASCII，10月号，474p.

▶SX-WINDOWプログラミング　最終回
SCSI装置に関する話題や連載中に作成した「FileCut.X」をマルチタスク対応に変更する。——吉野智興，C MAGAZINE，10月号，124-129pp.

ポケコン

PC-E500

▶ポケコン100M走
キーを素早く押してタイムを競う。——御茶目菜子，マイコンBASIC Magazine，10月号，122-123pp.

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
電撃王　主婦の友社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

QUESTION and ANSWER

Oh!X 質問箱

Q **なぜ，X68000PRO/PROIIはボードが使えないことが多いのですか。**

神奈川県　山本　勝(ほか多数)

A X68000PROシリーズはツインタワーのマシンのものとは少し基本設計が違います。最近のパワーアップハードのようにシビアなタイミングを要求するものではそういった違いが問題になってきているようです。

また，スロットが多い割に電源が弱く，結局SCSI2ボードの場合は電源容量が足りないことがたたったようです。SCSI2用のコントローラが非常に電力を食うので不安定になり，なんとか動いてもメモリが増設できない状態になるので実用にならないということらしいです。拡張ボードをどのスロットに入れるかでも安定度が変わってくるようです。

そのほか，不安定になる原因を究明しようにもPROシリーズに限って回路図が世に出ていないのでほぼお手上げ状態だとか。

あるマシンで動いても，製品によるばらつきが大きいようです。Xellent30PROではPROシリーズにも対応していますが，ここではなんと製品によってタイミングを3通りに変更できるように設計して対処しています。

Q **X68000に28800bpsなどの高速モデムをつなげても無駄と聞いたのですがなぜですか。**

東京都　菊池　生

A これはX68000の問題というよりも単に使用するソフト側の問題でしょう。

ここでいう28800bpsというのはモデムと電話回線を通したホストのモデムとのあいだの通信速度を表しています。モデムはパソコン側から送られた情報を圧縮して転送しますので，モデムとパソコンのあいだはモデム-モデム間の倍（57600bps）くらいで接続されているのが理想です。

しかし，通信ソフト（とドライバ）によってはパソコン-モデム間を19,200bpsでしか接続できないものもあります。こういった状態だとせっかくの高速モデムが生きてきません。確実に19200bpsは出る分，14400bpsのモデムをつないだときより高速にはなると思いますが，たいした恩恵はありません。

とりあえずパソコン-モデム間を34000bpsくらいで接続できる通信ソフトとドライバを使用されることをおすすめします。

なお，パソコン-モデム間の転送速度は倍くらいが理想と書きましたが，実際に高速になってうれしい部分（バイナリデータのダウンロードなど）ではほとんど圧縮はきいていませんので，28800bpsより大きければ体感速度はほとんど変わらないはずです。

Q **辞書データをテキストに変換したときに単語コードのような数値が出てきますがあれは意味はどういう意味なのでしょうか。**

岩手県　島村　吉雄

A 最初の2桁は品詞のコードです。これはだいたい表1のようになっています。

動詞は特に説明は必要ないでしょう。形容詞は名詞形によっていくつか種類があるようです。14番は名詞形の少ないものです。「大きい」などが14番ですが「め」や「げ」をつけたものは変換できません。15番は，「危ない」のように，「げ」をつけることができます。16番は「おもしろい」のように「げ」，「み」をつけることができます。17番は「新しい」のようなもので，「げ」，「め」をつけることができます。18番はデフォルトでは「強い」だけが登録されており，「げ」，「め」，「み」をつけることができます。大半の形容詞は14番で登録されています。

続いて名詞です。20番の形動名詞は名詞の後ろに「な」を伴う用法が可能な名詞だと思っておいてください。21番はテキスト出力では「名詞」とはなっていますが，形動名詞に近いもので，「たる」をつけることのできる形動名詞というところでしょう。22番のサ変名詞は「する」をつける用法のある名詞ですから判別は簡単ですね。これらに当てはまらないものは24番の名詞になります。

「物の名称」や「団体名」などがどの程度機能しているかは謎なところです。地名，人名もあまり当てにはなりませんし。

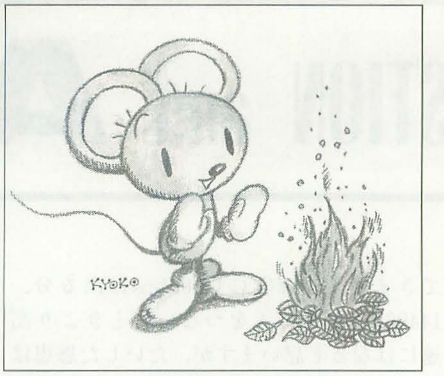
後ろに続く数値は学習コードです。通常3桁で大きい数値のものほど優先して変換されます。サ変動詞，接頭語，接尾語などでは5桁のこともあります。

接頭語/接尾語に関しては品詞の直後2桁が学習コードで，ここが00以外のだとほかの品詞よりも優先して候補になります。残りは副品詞コードというべきものです。本来は学習コード3桁，副品詞2桁なのに辞書メンテの不都合でこう出力されているようです。　　　　（中野　修一）

表1　ASK3の品詞コード

01	カ行五段活用動詞	22	サ変名詞
02	ガ行五段活用動詞	23	ザ変名詞
03	サ行五段活用動詞	24	名詞
04	タ行五段活用動詞	26	地名
05	ナ行五段活用動詞	27	人名（姓）
06	ハ行五段活用動詞	28	人名（名）
07	マ行五段活用動詞	29	団体名
08	ラ行五段活用動詞	30	物の名称
09	ワ行五段活用動詞	31	数詞
10	サ行変格活用動詞	32	感動詞
11	ザ行変格活用動詞	33	接続詞
12	カ行変格活用動詞	34〜37	副詞
13	上下一段活用動詞	38	連体詞
14〜18	形容詞	39	慣用句
19	形容動詞	40	単漢字
20	形動名詞	45	接頭語
21	名詞	50	接尾語

STUDIO X ON AIR



FROM READERS TO THE EDITOR

冬の足音がヒタヒタと忍び寄ってきました。冷たく澄みきった空気というのも気持ちがいいものですが，今年の冬は，夏場の猛暑の反動で激寒にならないといいな。お願いです。寒気団さん，あまりがんばらないでね。

◆X68030(33MHz改)とSCSI-2ボードを使ってデータを垂れ流す程度のことで，特集が組まれているのは残念だ。X68000ならまだしも，X68030を使うからには，なんらかの圧縮アルゴリズム(それほど高度でなくてもよい)をもう一歩踏み込んで考案or提案すれば，内容も少しは深くなるのではないだろうか。Oh!Xを値上げしておきながら，この程度の内容しかないのなら，長年買い続けていたけれど購読をやめようかと本気で考えた。　新子　弘康(24)大阪府

◆9月号を読んで，きちんとした基本技術の大切さを痛感した(VRAMのゴミっていったい……)。最近つくづく思うのだが，どんなクズハードでもその機能をフルに使用したソフトは，みんな輝いている。だから私はX68000版の「スペースハリアー」がいまでも大好きだし(中野氏の意見に賛成)，スーパーファミコンの「ファイナルファンタジー」シリーズの曲は，CD-DAで演奏されている曲に負けないくらい素晴らしいと思う(特に音量変化がうまい)。

折坂　信春(25)大阪府

◆もうこうなったらOh!Xでハードディスクアニメーションのためのきちんとしたフォーマットを決めてしまいましょう。シャープは当てにならんし。　佐井川　泰治(23)東京都

◆「Mach-2」の発売により，X68000もまたひとつ進歩しましたね。今年はソフトウェア方面はさびしいけど，ハード面では「AWEASOME-X」や九十九電機製16Mバイトメモリや高速通信ボードなどの充実が見られますね。僕自身も早く皆さんになにか提供できる力をつけたいと考えています。　大道　顕二郎(20)岩手県

◆New Xに求める性能。CPU……PowerPC最新型(シャープがアップルと手を組んだのはこれのために決まっている！　かも)。画面周り……1024×1024ドット24ビットフルカラーくらいでプレイステーション並に動いてくれたらすごく嬉しい。音源周り……SY77!　最低でもB700の音源部をそのままで。波形はFDかなにかで自由に書き換え可能(ああ，SY77のスーパーFM音源をビシバシダイレクトにアクセスしたい)。一応，GM規格配列のボイスFDをOSと一緒につけてもらえればなおよし。CD-ROMは，つけるくらいだったら上記の機能を実現させてほしい気がする。でも実現したとして，はたしていくらになるのやら。でも夢は大きく……。

松嶋　登(20)神奈川県

◆次期Xについて考えてみると，OSはUNIX(BSD系)以外に考えられないですよね。CPUはR4000かPowerPCか，はたまたAlphaかSparcか。DSPは当然シャープ製のカスタムチップを用意。これが，24ビットのVRAMとか16ビットのリニアPCMを駆使することに……。そしてX Window Systemをシャープ流にアレンジしたものをユーザーインタフェイスにして，これはもうXシリーズ第5世代ということで機種名は，X5で決まりだね。　佐渡　詩郎(19)石川県

◆「シャープは高速CPU搭載のパソコンの2次キャッシュメモリ用として(以下略)……」トランジスタ技術9月号より。この程度のニュースで反応している自分が悲しい。

及川　恒平(20)東京都

◆最近無限ループ状態になって困っています。Macintoshが欲しいなあ。しかし，CHRPマシンが出るまで待ったほうがいいかな→だったらWindowsマシン(DOS/V)がいいかな。しかし使いにくそうだし→だったら新Xを待つかな。しかしいつ出るかわからんし(笑)。

田川　和義(21)岐阜県

◆9月号のプレゼントの「Inside PowerPC」はいったい……？(ちょっと期待)

竹本　郁馬(24)神奈川県

ここ最近盛り上がっている新機種の噂。はやく真実を知らせてほしいものです。ちなみにプレゼントのシロモノは，ご想像にお任せします(みんなわかってくれるよね)。

◆「X68000ゲーム年代記」は，いまだからいえる裏話が聞けてよかった。これからも裏話をどんどん暴露していって！　それにしても昔はよかったねえ(じじい)。　加藤　和人(19)愛媛県

◆「X68000ゲーム年代記」は，懐かしくて面白かった。連載，楽しみにしています。

是枝　浩行(35)神奈川県

◆THE SOFTOUCH REVIVALはいい企画だと思います。過去の名作ソフトの隠れた遊び方や裏技，そして新しい発見などなど，個人で気づかないことについて再レビューしていただき，骨のズイまで遊びつくせるようにしてほしいと思います。

後迫　浩一(34)神奈川県

◆9月号からTHE SOFTOUCHの内容が変わりましたね。「X68000ゲーム年代記」はとてもいい企画だと思います。しかし「リバイバルレビュー」に関しては，少し内容に疑問があります。現在取り上げたとしても，読者のほとんどが買いにいくことのできないゲームを，カラー4ページ使って紹介することはないと思います。個人的には，このコーナーではTAKERUで買えるものを中心に扱うといいと思います。

青木　英郎(20)大阪府

◆「サイバリオン」の記事にグッときました。私はこのゲームをプレイするときには，数年前に友人から2,000円で買ったアスキースティックを使っています。グニグニしているのが割とイイ感じです。せめてメガバリオンをしばいてみたい今日この頃です。藤本　将景(25)高知県

読者の皆さんに認めてもらえるように，新しいTHE SOFTOUCHはがんばっていきます。



◀岡村　直也　京都府
SHARPな人生か(どんな人生か考えている)……SHARPな(もっと考えている)……「コンビニ行こっと」(どうやらお腹が空いたようだ)。



◀渡辺　宏幸　埼玉県
大人の恋にしびれちゃう～ってか。きっとこの友人は，アルフェイムを自分の高校生活と重ね合わせながらプレイできるんでしょうね。

ます。
◆そっかあ。「HAYA OH」ってX68000版のオリジナルのボスだったんだ。私はずっとSEGA mkIII版のオリジナルだと信じてたんだけど。そっか，そうなんだあ(スゲー嫌味)。

久米　豊信(27)群馬県

このほかにも，間違いを突っ込んでくれたハガキが数枚……ごめんなさいね。

◆某アスキーの某ファミ通別冊の某ネオジオ通信第2号108ページ右上のイラストを見てみよう。おやあ？　ま，それはともかく僕は「サイバリオン」でグラフィックは気になりませんでした。一番のキモはゲームそのものの独自性ですし。ところでアンケートハガキの「今月号の編集について」は誤植なんでしょうか。これ，本気で聞かれたら困っちゃうな。「方向性はベスト」とか答えましょうか。高橋　明(25)東京都
◆「今月号の編集について」というのは，どういうことなのでしょうか。編集の人が変わった(代わった？)というのでしょうか(笑)。

新谷　貴幸(20)埼玉県

結論からいってしまえば誤植です。でも割と真面目に答えてくれる人が多かったのは，ちょとだけ嬉しかったかな。

◆ウヒヒッ，ついにハードディスクをパワーアップしました。ドライブはConnerのCFP-1080Sです。1030Mバイトで認識します。速いです。CONFIG.SYSが流れます。でも，以前使っていたハードディスクが日本製だったので，固定ネジが合わずにカタカタ揺れます。不安です。地震がきたら間違いなく落ちます。しかたないのでとりあえずガムテープ(!)で固定していますが，見た目があまりよろしくないので，秋葉原にネジを買いに行ってきます。

久保田　智久(20)群馬県

いってらっしゃ〜い。でも，浮気はダメよ。

◆ついにX68000XVIのドライブ0が壊れた。「書き込み不能」「プロテクトがかかっています」「ディスクが入っていません」「フォーマットできません」。なにを脈絡のないことをいっているんだ！　どうしたんだX68000 XVI〜。ちなみにX68030は今日も元気に動いている。

市川　博基(20)愛知県

このような症状が出たら，即座に入院させてあげましょうね。あまり無理やり働かせるともっとすねちゃいますから。

◆DSPの連載がハード，ソフトともに本格的なものになりそうです。こうなると問題はDSPボードを買うためのお金を貯める間にDSPボードの入手が可能であるかどうか，ということでしょうか。　　下川　将紀(24)東京都

実機があるとないとでは，連載の楽しさも格段に違ってきます。がんばって手に入れてね。

◆9月号の「響子inCGわ〜るど」を読んで，ジャズバンドのストリートパフォーマンスを，パイプをくわえながら静かに眺めていた見事なヒゲのホームレスのおじいさんを思い出しました。CGもさることながら，江口響子さんのエッセイは，毎月本当に楽しく読ませてもらっています。「猫とコンピュータ」みたいに，この連載も単行本になると嬉しいな。溝田　幸弘(25)大阪府

毎回なにか感じさせてくれる江口さんのエッセイ。今月も溝口さんのお気に召しましたか？

◆カセットデッキの調子が悪くなったので(音質が急に低下した)，かなり前，STUDIO Xに載っていた“アジマス比の調整”というのに手を出してしまった。危険だとも書いてあったので怖かったけど，なんとかバッチリ復活した。こういうものに限らず，Oh!Xの記事には，何年か経ってから役立つものが非常に多い。自分が遅れているだけかもしれないけど。

吉田　努(23)大阪府

生活の知恵袋としての利用もできるOh!X。毎月18日発売です。

◆キーボードが壊れました。ENTERキーが引っ込んだまま，引っかかって出てこないのです。しかたないので油を差してやったら(滑らかにキーが出てくるようにと思って)直りました。が，フィーリングが変わってしまいました。スカスカです。わ〜い。　須佐　英之(21)千葉県

なんて強引な治療法。副作用が出ないことを祈っています。

◆やっと増設した3.5インチフロッピーディスクドライブが動くようになった。いままで動かなかった原因が，電源の容量不足でディスクのモーターは回るが，ヘッドのステッピングモーターが動かなかったのであった。しかし，最近はMOでデータのやりとりをするので，あんまり意味がないのが悲しいところ。

浮田　衛(24)千葉県

自分で使わないのなら必要な友達に貸してあげるといいかもね。

◆「PC-TM151」に注目しているのですが，9月



◀千葉　茂夫　埼玉県
パッドが壊れるのが先か，千葉さんの手が壊れるのが先か……それともBEAT君の修理が先送りになるだけだろうか。ちょっと心配。



◀佐藤　貴是　神奈川県
うんにゃ。体育祭といえばチアガールでしょう。学ランを着た女の子の応援団というのもなかなかいいけどね。

新機種に期待する機能は？　その1

9月号アンケートハガキにある「新機種に期待する機能はなんですか？」の項目をまとめてみました。読者の皆さんが，どのようなマシンを望んでいるのか見ていきましょう。

・怪しくないものは不可
・誰もがクリエイターになれるような機能
・30Mバイト/secの転送可能なバスとDAIと640×480ドット24ビットフルカラーで，24ビット96kHzサンプリングのA/D，D/Aコンバータ
・現行のX68000より上ならOK。そんなことよりデザインです
・X68000なみの感動，X68000並の使いよさ，X68000とは思えないサポートのよさ(笑)
・ホームセキュリティ機能。PD。そして，ビデオCD規格
・夢！
・パソコン本体にテレビチューナーを内蔵していて，ハイビジョンやワイド放送の画像取り込み放題
・X68000との互換性はどうでもいいから，個性のあるパソコンにしてもらいたい。AT互換機やMacintosh互換機とかは論外。新SX-WINDOWなんてあるといいかも
・CG機能。DōGA並のアニメーションがさくっと動くと気持ちいいかもしれない
・スプライト，MIDI(インタフェイス)，マンハッタンシェイプ，赤外線センサー，オーディオIN/OUT，ビデオS入出力端子，インタラプトスイッチ
・ポリゴン，MPEGエンコーダ，デコーダ，赤外線通信
・ニューロコンピュータとか超強力バスとか並列アーキテクチャの採用だとか。なにかひとつでもいいから目立ってほしい
・フラットなアドレス空間と直接叩けるハード類。高速なI/Oバス，拡張性，PPC-604
・RICS CPUであること。オリジナルアーキテクチャであること。Windowsが走らないこと。UNIXが走ること。スプライトがあること
・標準では最低限のスペックでいいから，かわりにどの用途でも手軽にハイクオリティを得られるような拡張性を考えてあり，さらにMIDIなどのインタフェイスも標準で付属していること。いちばん願いたいのは，シャープが強力にバックアップしてくれること
・超速いマシンを作って，MacOS，Windowsをエミュレーションできようにする。MacintoshやDOS/Vの部品を使えるようにする

号の九十九電機の広告を読むと「横端の表示が多少歪みます」とある。うーん不安だ。改善策はあるのでしょうか。ということで10月号の新製品情報を期待しています。
林　周秀(19)千葉県

確かに歪みがありますが，ちゃんと調整ができるので問題はありません。10月号のレポートでもその点は解説されていましたので，安心して購入できたかな。

◆先日「Easydraw」を買いました。けど忙しくて(＝疲れて)インストールすらしてません。ああ，なんたることよ。金のある社会人でも暇がない……昔はよかった，かな？　次はスキャナとフルカラープリンタとソフトで年賀状ですか。しかし，さすがにこれだけのものだと先立つものが……。
及川　雄也(24)宮城県

社会人の強〜い味方，冬のボーナス一括払いがあるじゃないですか。

◆会社でワークステーションのモニタ画面が大揺れしており，いかにも目を悪くしそうだったので電磁シールド(金属の板でモニタを囲んで画面の揺れを防ぐもの)を購入しました。最初，たいした効果はないだろうなとなめてかかっていたのですが，揺れがピタリと止まったのにはビックリ(細かな揺れは残るが)。強力な電磁波(窓の外を電線が走っている)をこうもあっさりカットしてくれるとはねえ。ちなみにこのシールド，4枚の折れ曲がった金属版だけだというのに5,6万円します。まあ，ちゃんと効果があったことだし，視力悪化を考えれば安いもの……と考えられますが。
渡辺　久孝(28)大阪府

ということで，電磁波にお悩みの方はディスプレイの周りに厚めの鉄板を張り巡らせましょう。材料さえ調達できればとおっても安上がり。成功レポートをお待ちしています。

◆X68000 PROII買いました(グレー，新品)。主な目的はマシン語の勉強です(もちろんC言語もかじりたい)。ただいま留年中で1年のうち345日は休日なので，幸いにも(?)時間はたっぷり……。幸せな1年が過ごせそうです。
土田　敬生(23)大阪府

好きなことをやっていると，時間は速く流れるものです。うかうかしていると1年間なんてあっという間ですよ。がんばってくださいね。

◆9月号ではさっそくのアドバイス，たいへんありがとうございました。WB音源完成の晩には，写真を添えてご報告させていただきます(いつのことやら)。ところで，現在X68000Compact XVIを買っても付属のSX-WINDOWはやはりver.2.なのでしょうか。む〜ん，オープンプライスの甘い罠ですね。
藤田　和久(32)富山県

そういえば，最近買ったX68000 Compact XVIのシステムはver.2.0だったなあ。有料でもいいからバージョンアップサービスがあるといいのにね。

◆9月号122ページのコメントを書いた編集さんへ。今年の春卒業した僕の高校の先輩は，ダンプリストが読めます。彼は周囲にさんざんいわれながらも，ハンドアセンブルを続けました。こんな時代にも珍しい人はいるものです。
小川　貴也(16)千葉県

その先輩は，ダンプリストは16(もしくは32)バイト単位で読みやすいんだ，なんてことをいっていませんでしたか？

◆この前，某出版社が出している本を買った。しかし，付録のディスクがうちのX68000 XVIには認識してもらえなかった。父のX68000 PROでもだめだった。そんなわけで，出版社に電話をして，ディスクを送ることになりました。ところが，送る前におもむろにドライブに入れてみると……動いているではないか。
飯田　雅代(18)東京都

慌てて訂正の電話を入れたか，そのまま知らん顔してディスクを送付したかは，飯田さんのみぞ知る。

◆現在のバイトは会社に泊まることが多い。で，わかったことは，人間どこにでも寝られるということ。床で寝る。イスで寝る。階段で寝る……ああ，布団で眠りたい。
中村　健(25)埼玉県

こうしてアウトサイダーな人間が作られていくのですね。

◆3時に「ぷよまん」が出ました(会社で)。出張のお土産だったらしいのですが，まさか仕事中に「ぷよぷよ」に出会うとは思いませんでした……怪しい。
岩瀬　貴代美(23)福岡県

僕も編集部へのお土産に買ってきたのですが，見かけの割には「ぷよまん」っておいしいですよね。結局，買ってきた本人が一番食べていたような……。

◆しばらく(1年ほど)X68000と離れた生活を送っていて(受験だった)，X68000 CompactXVIちゃんの前で“さあって，やるぞ！”と4月から気合を入れたのにもう9月。結局なにもできなかった。芸術の秋，X68000 CompactXVIちゃん，一緒に夜更かししよーね。
松尾　絢(18)東京都

あんまり待たせるのも悪いけど，うちのX68030みたいに毎日酷使していると「ええかげん寝かさんかい！」と怒られそう。

◆ふと思ったのですが，Oh!X編集部の皆さんは，全員タッチタイピングができる人たちばかりですか？　やはり，パソコンを持っているなら，タッチタイピングは当たり前なのでしょうか。
片山　明義(17)奈良県

できる人もいればできない人もいます。僕なんか，パソコンを使い始めてから10年以上経ちますが，いまだにめちゃくちゃなタイピングです。しかし，できるにこしたことはないので，「あ，できなくてもいいんだ」と思わずにできるように挑戦してみては。

◆とにかく努力が嫌いな私が，今年生まれて初



▲岩瀬　貴代美　福岡県
スピード感はいいけど，もうちょっと線と墨にメリハリをつけるとメカメカしい感じになって迫力が増すと思うぞ。

新機種に期待する機能は？　その2

・メモリ64MバイトでMPEGが標準ね！　あとはPDS(プロセッサダイレクトスロット)！
・変化する力と安心感
・アスペクト比1:1で1670万色同時発色で，CPUは68系のものを。CD-ROMは当然だが，この際MIDIも内蔵してほしい
・人の言葉を理解するニューロチップを使った入力装置
・2048×2048ドット(15ビットカラー)や1024×1024ドット(24ビットカラー)といったグラフィックモード。そして，フレームバッファによるスプライトおよびポリゴン
・開発環境をデフォルトで！
・ホームオートメーション(家電をすべて扱えるようなもの)
・グラフィック。特に広い画面はどうしてもほしいです。もちろん，1:1のアスペクト比で。GUIは，SX-WINDOWの流れをくんでほしいです
・圧倒的に速いマザーボードとオープンアーキテクチャ。スプライトと6倍速CD-ROM。Windows95などいらん
・CPUがグレードアップしても(クロックアップしても)足を引っ張らない拡張バスとVRAM(ローカルバスしかないか)
・DOOM, DOOMII, NASCAR RACINGがプレイできればそれでいい
・完璧なマルチタスク環境を作れる機能
・パワーアップしたスプライト機能。X68000との互換性
・高解像度，PowerPC，MacOS互換，メモリ上限大，5年以上もつCPUパワー
・半年はハマるゲームがついてくること。8年間仕様が変わらないくらいのスペックであること
・ゲーム機に負けないAV機能
・ゲームを作成しやすいOS
・動画がガリガリいじれるパワー
・通信機能の充実
・自分で手軽にプログラムが組める環境
・ビデオ出力を標準で！　DOS/Vのエミュレーション
・ネットワーク，ザウルスとのリンク，他機種のデータ利用
・本当の意味でのマルチメディア機。5年後に標準となるべきスペックを詰め込んだ機種
・ソフトサポートの充実
・Windows機に流されない独自のアーキテクチャ。音楽屋の私としては，音源周りを充実させてほしい(PCM8音くらい)

めて自分から目標を立て，実行に移しました。その今年，なぜか世間では大きな事件が続いています。「自分に似合わないことはするな，という神のお告げかな」と思う一方，「大丈夫，まだ沖縄に雪が降っていない」と自分にいい聞かせてがんばっています。　藤原　彰人(25)岡山県

ということで今年の冬がとっても楽しみ……かな？

◆7月の大雨による災害で，月一，二度しか山奥(勤務地)へ登らない。その山奥での仕事がかなり少なく(まったくないわけではない)，暇な時間にOh!Xを読んだり，ローテクのパターンを書いたりしている。しかし，月一，二度しか愛機に触れないのが悲しい。

黒田　博明(26)富山県

このまま黒田さんの気持ちがX68000から遠ざかってしまうのか，それとも中古のX68000を手に入れてしまうのか。Oh!Xは静かに見守っています。

◆皆さん，クーラー病になっていませんか。現在，9月6日ですが，テレビで注意するようにといっていました。最近は，クーラーよりも自然の風のほうが涼しいように感じます。体には気をつけてがんばっていきましょう。

弘山　和広(20)東京都

確かに今年の夏は異常に暑く，エアコンをガンガンに回していましたね。おかげで外気との温度差によってずいぶんと苦労しましたよ。さらに木造のオンボロアパートに住んでいるため，電気代が20,000円(熱効率が悪く，エアコン24時間フル稼働)もかかってしまって，財布もピーピーです。

◆最近，自宅でラップ現象がよくある。悪い霊のような気もするが，明け方ディスプレイ(ひと晩ついていない)なんかがなると不気味。ラップ音なんかどうでもいいけど，ディスプレイが壊れるのは怖いですから。

滝本　哲之(22)埼玉県

体に不調を訴えるような症状がなければ，たぶんそれは霊現象でもなんでもないはずです。原因は，明け方，大気が冷えて物体が縮むときに発する音だと思うのですが。きっと，なにも心配することはないですよ。

◆はっきりいって，私は自分のパソコン歴がわからない。私が初めてパソコンに触れたのは，幼稚園のとき(MZ-80BかMZ-2200かX1cだと思われる)。父がどこかの雑誌から打ち込んだと思われるゲームをしていたような気がする。それからゲーム機くらいしかいじらない期間が長く続き，中学2年生のときにFM TOWNSを買ってもらい，パソコンに深入りしたのは今年からだ。さて，私のパソコン歴はいったい何年だ。

佐藤　浩人(16)福島県

特にこれといった基準はないので，あまり悩まずに決めていいと思いますよ。具体的には，パソコンを使い始めてある程度継続していた期間からカウントすればいいんじゃないかな。佐藤さんの場合は，中学2年生の頃からとしてパソコン歴3年というのがいちばんいいのかな。

◆パソコンっていったいなんのために使うのでしょう(不毛か？)。目的のための手段であって，手段が目的になってほしくないなあ。特にNew Xは。　平島　俊英(17)埼玉県

僕は自分のやりたいことを実現するためにパソコンを使っています。要は使う人の心がけひとつで決まると思うのですが。

◆最近のビデオはどんな機能がついているのかと思い家電販売店へいってみた。すると1本で表，裏番組をチェックできるビデオがあるとは……どこのメーカーだ，と本体を見ると「SHARP」の文字が。店頭マニュアルで確認したら，そのモードで録画すると，ほかのVHSデッキでの再生が不可能になるとか。これも目のつけどころが……ですか？　菅谷　英明(24)兵庫県

間違いなく目のつけどころが……ですよ。

◆会社の先輩に「お前さんはゲーマーだな」といわれました。その先輩には，ゲームをやっているところを見られた覚えはないのですが。やっぱり「ドイツ人ジャーマン」を口ずさみながらワープロを打っていたのがいけないんでしょうか。　宮野　文武(22)栃木県

ほかにも道を歩くとついついマス目どおりに動いてしまうとか，ボタンを連射するときついつい2本指で叩いてしまうとか，ちょっとしたクセを見抜かれたのかも。

◆私は最近ネコを飼い始めました。しかも子ネコ！　ミルク代，ほ乳ビン，予防接種といろいろとお金がかかりますが，かわいい～！　もうじき自分の子供ができるのですが……同じような状態になるんだろうなあ。まあ，とりあえずは子ネコで……。　谷口　博一(29)京都府

しっかり予行演習(?)しておこう，というところですか。

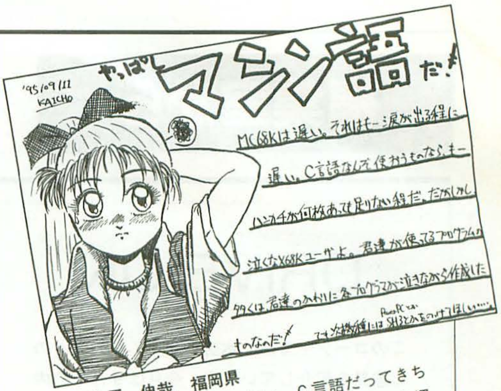
◆12月号に付録ディスクを予定しているみたいですね。通信ができる環境にないのでたいへん嬉しく思います。ただ，「○○というフリーソフトが必要」とされてしまうと，かなりつらいものがあるので極力避けてください。夏は出費がかさみましたが，1にOh!X，2に電脳倶楽部，3，4がなくて，5に食費とがんばっています。

松鶴　正博(22)静岡県

12月号の付録ディスクは豪華絢爛，本邦初，大盤振舞の2枚組との噂が……期待していてください。あ，そうそう食費を削るのはほどほどにしましょう。理由は，次のハガキを読んでね。

◆8月末に退院しました。3カ月弱の入院で体力(筋力？)がすご～く落ちてしまい，定時出社/退社するだけでも疲れてしまうのでまいっています。やっぱり入院なんかするものじゃないですね。健康が一番。ちなみに病名をばらすと肺結核……。睡眠時間と食費を削るとやばいです。注意しましょう。　鈴木　道明(26)埼玉県

病気になってからわかる健康のありがたさ。身に覚えのある人は気をつけましょう。病気になってからじゃ遅いですからね。



▲奈良原　伸哉　福岡県
速い遅いはプログラマしだい。C言語だってきちんと使えばかなりのことができるはずです。大切なのはプログラミングセンスでしょう。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフト・コンピュータ本体の売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★ローランドの音源モジュール「CM-32L」を12,000円で売ります(送料込み)。箱，説明書，付属品あり。買ってくれた方には，「スーパーストリートファイターIIX」のテレカをつけます。連絡は往復ハガキでお願いします。〒018-01 秋田県由利郡象潟町浜山4-16　三浦　栄悦(27)

★アイテック製ハードディスク「ITX680」(黒)を20,000円(箱，説明書，ケーブルあり)，アイ・オー・データ機器製増設メモリボード「PIO-6BE4」(拡張スロット用4Mバイト)を20,000円(箱，説明書あり)，NEC製熱転写プリンタ「PC-PR801」を50,000円(インク，用紙，箱，説明書あり)でそれぞれ売ります。連絡は往復ハガキでお願いします。〒178　東京都練馬区大泉学園町7-23-21　大泉ハウス201　秋山　和徳

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は9月号の内容に関するレポートです。

●私のいまの動画再録に対する姿勢は，昔のパソコン通信に対する祝一平氏のそれに等しいです。実用的な使い方なんて，DōGAが最高峰くらいで，「マイコンBASIC Magazine」の付録CD-ROMを読むのがせいぜいでしょう。そう，動画はCD-ROMが大前提でもあります。CD-ROMから読むのではなくて動画をやりたいなら，まだそれなりに苦労があって当然(かつ，それを乗り越えること自体を楽しみとするのが当然)という時期だと考えていますので，29ページの「これでは編集作業に手間がかけられるはずもない」という一文や，「アニメーションの現状」にあるようなメーカーに頼る姿勢(そんなに需要があるとは思えないのに……)には，反感を覚えてしまいます。

しかも，「ビデオ作品制作という目的もよいが，それ以外にもアニメーションというものは使用価値のあるものである」というのは大嘘でしょう。最初に書いたようにCD-ROMで供給されると実用的なものもありますが，アマチュア発アマチュア行きの動画で，ビデオ作品制作以外となれば，9割が余興にしかならないでしょう。それも，どこぞのテレビアニメを見て喜ぶのがそのほとんどでしょう。

鈴木　朝夫(21)　X68000，MZ-1500，X1turboZ，PC-9801RA，PC-88VA2，PC-6601，MSXturboR，FM-77AV，ZX-81　神奈川県

●無圧縮でアニメーションをやろうとはSCSIの速度が上がっただけでずいぶんすごいことができてしまうのですね。個人的にこうした力技はあまり好きではないのですが，「力技ができない」のと「力技を使わない」というのは違いますから……。

それにしても，X68000のボトルネックになっている部分のなんと多いことでしょう。CPUの処理速度も，グラフィックの表示能力も，スプライトの表示数も，なにからなにまで足りません。8年以上前のスペックですからしかたないのかもしれませんが。とりあえず，CPUは040turboで，SCSIはMach-2でなんとかなりました。しかし，ほかの部分はいつになったら解消されるのでしょうか。

特集とは直接関係ないのですが，仕事の関係もあって，最新ゲーム機で遊びまくっています。そして思うのはアニメーションの汚さです。特集のなかで触れられているように，SEGA SATURNなどのアニメはシネパックがよく使われています。そしてシネパックは画質をかなり犠牲にした方法です。離れて見れば汚さが目立たないのでしょうが，ゲームのようにかなり近くで見る場合は最悪です。パソコンのようにドットが細かければ，それほど気にならないかもしれません。しかし，家庭用ゲーム機は最新ゲーム機といえどもドットはかなり粗いんです。まともに見られるものは昔ながらの目パチロパクなどを使った「ゆみみみっくすREMIX」だったりします。

中村　健(25)　X68030，X68000 ACE-HD，PC-386GS，MSX2＋　埼玉県

●新連載の「Digital Signal Processing」ですが，私はDSPボードとはなんだろうと思っているので楽しみです。それに，DSPそのものの基本知識と基本動作を理解している人は少ないと思うので，これからが楽しみな連載でしょう。

壁谷　善嗣(36)　X68000 EXPERT，PC-9821As，PC-9801NS/E　宮城県

●新しく始まった「X68000ゲーム年代記」はよかったと思います。いままでアンケートハガキに「X68000のゲームカタログを別冊で！」と書き続けていた僕にとって，まさに待望の記事でした。おそらく「新作ゲームが出ないから過去のゲームを取り上げている」と思われそうですが，そういった消極的な考え方をせず，「いま遊んでもこんなに楽しいんだよ」ということを積極的にアピールするような記事をこれから期待します。ただ，残念なのは「こんなに面白そうなゲームが過去にあったのか」という発見があっても，いまでは入手困難なことです。

北浦　暁光(21)　X68000 XVI-HD，X1turboZIII，PC-8801mkIIFR　東京都

●「MISSILE SYSTEM」が面白いですね。操作系のハンデを克服すべくフルキーオペレーションという工夫を思いつく，まさにピコピコ精神あふれるゲームではありませんか。これでバックにキートップの絵を書いておけば，そのままタイプ練習になりそうですね。偏った練習になってしまうかもしれませんが……。ハードの導入に頼らない解決というものは，やはり気持ちのいいものです。

石田　伯仁(22)　X68030，MZ-731，PC-8801mkIIMR，PC-E200　神奈川県

●プロによるゲームソフトが出なくなったX68000(パソコン全体という話も……)。そんななかで注目すべきものは個人レベルでのソフト。TAKERUがそういったものを取り込む姿勢も素晴らしい。また，それを紹介していく記事もこれまた素晴らしいと思う。

今回は他機種ではほとんどみられないアクションゲームの楽しい作品がずらりと並んだ。これからもゲーム制作者にはがんばってもらいたいものである。そして「THE USER'S WORKS in TAKERU」に期待する。

三隅　信幸(22)　X1turboZII　広島県

●9月号でいちばん気になった記事は「OCR.X」です。オーディオIN端子にこんな使い方があるなんて……。私がこのアプリケーションに直接なにか恩恵を受けることは少ないと思いますが，なんか「イイ」です，これ。

あと，もうひとつ気になったのが，「(で)のショートプロぱーてぃ」の見出しです。「「自分で作れ」の精神を見た！」最近プログラムをなにも作っていなかった私にとってこの言葉は心に効きました。しかしながら，現在は仕事の関係でパソコンが触れない環境にいるものですから……。

大上　幸宏(22)　X68000 PROII　鹿児島県

ごめんなさいのコーナー

9月号　Oh!X LIVE in'95

P.66　WAIT FOR SLEEP

作者の名前が間違っていました。正しくは千喜良和明さんです。千喜良さんには大変ご迷惑をお掛けいたしました。

10月号　(で)のショートプロぱーてぃ

P.64　MARUCO.FNC

「●St, ed」のなかでst<edとあるのはst>edの間違いです。また「MARUCO.FNC」を作る手順は，「MARUCO.S」をアセンブル，リンクして「MARUCO.X」を作成し，リネームすれば結構です。あと，インクルードライブラリの新旧によっては311，319行で「undefined symbol error」が出る恐れがあります。エラーが出たときはシンボル「_SUPER」を「__SUPER」に変更してください。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

はりきれば そこには 未知の世界が……

▶今年の1月号に続き，2回目のCD-ROMの特集です。X68000ユーザーの間でもだんだんCD-ROMドライブを所有するユーザーが増えてきました。それとともに，書籍でもCD-ROMつきのX68000関係の本が出てきています(当社だけという話もありますが)。ただ，X68000は市販ソフトの供給もままならない状況にあり，CD-ROM用のソフトが供給されているわけではありません。それでは，なぜCD-ROMを使う必要があるのでしょうか？

CD-ROMはデータを供給するメディアとしてはかなりポピュラーなものです。そして多くの機種で使用されるデータがX68000でも利用可能なのです。これを放っておく手はないでしょう。もちろん，CD-ROMをもってくればそれだけでOKというものではありません。そのデータを利用するためにプログラムを作ったりデータを解析したりする必要もあるでしょう。今月の特集を参考に自分でもなにかやってみる気になっていただければ嬉しいのですが。

▶今月号では久々に「X68000マシン語プログラミング」が復活しました。前回の登場が1994年7月号ですから，1年以上のブランクがあります。最近読者になられた方だと初めて読むという方もいるかもしれません。マシン語でプログラミングをしたいという方には参考になる点が多いと思いますので，何度もプログラムを見ながら読みかえしてみてください。来月号でもお届けする予定ですが，著者がスランプを脱しきったわけではないので，登場するよう祈っていてください。

▶来月号では今年3回目の付録ディスクを予定しています。そして，まだ決定ではないのですが，2枚組になる可能性があります。その際，通常の付録ディスク時よりも値段がアップということになりそうです。内容の充実をはかるということで，ご勘弁ください。

▶今月は，著者多忙のため「こちらシステムX探偵事務所」「Digital Signal Processing」などがお休みです。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ(マシン語の場合)に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶ひと夏放置しておいたお米からまたもや蛾が大量発生してしまった！ 前回はブレンド米ブームだったうえ，被害の量も少なかったのでどうってことはなかったのだが，今回は8キロ近くも残っていたので大打撃である。ちなみに，その米を洗う際に幼虫&成虫を丹念に除去して炊きあげたとしても顎が外れるほどマズい！ (H)

▶暇だったので漫画でも買おうと思った。アニメ化されているという「ふしぎ遊戯」を買った。へー，面白いっすね，これ。久しぶりに満足できる漫画に出会えたような(大袈裟かな)。13巻で終わっていればすごくまとまった作品だと思うのに，まだ連載が続いているというのが残念。ドラゴンボールのようにならなければいいが。 (けんと)

▶「C MAGAZINE」のお蔭で，DJGPP(DOSのg++)の環境が整った。いつのまにかg++が身についていた。X68000とPC98で共通ソースが組めるようになったと思ったら，X68000ではメモリが足りないと動かない。くそう。立場逆転。まあ，OSのすみずみ(ってほどでもない)までである程度わかってるX68000は，私にはよい玩具かもしれないが……。 (瀧)

▶デジタルビデオがちょっと気になる。いままではビデオにはあまり興味がなかった。それというのも，「しょせんはアナログじゃん」という考えがあったからだ。しかし，デジタルとなると話は違う。いずれ据え置きも出て，パソコンなんかともデジタルでつながったりするんだろうな。一部のマニアだけの機器になりそうな気がしないでもないが。 (I.K)

▶今年はいろいろと激動の年になってしまい，1995年は長く記憶に留めることになりそうだ。人は多くのものを時間の中に捨てたり忘れたりしていくが，失うことを恐れずに新しいものを産み出していきたいものだ。なくなるといえば大阪のグリコの看板だが，10回以上大阪に旅行しておきながら見ずじまい。残念無念後悔山積だな，うんうん。 (八)

▶世間では第3次声優ブームらしいが，先日初めてドラマCDの収録というものに参加した(といってもエキストラだが)。しかし，ただの笑い声も満足にできず，ガヤ収録の際には言葉が続かず黙ってしまうこともしばしば。やはり声優さんってすごいんだなあ，と認識を新たにした1日でした。えっ，そのCDのタイトル？ まあいいじゃないですか。 (哲)

▶ただいまー。新婚旅行先のフィジーから帰ってきました。パソコンショップもゲーセンもなかったです。つまらん。ところで式の日に台風上陸という素晴らしい経験をした私ですが，なんと新婚旅行でも8日間中なんと3日間は暴風雨というこれまたグレートフルな体験でした(乾季だとゆーのに)。私がいったい何をした……人生の心得違いか？ (で)

▶セガの新作バイクゲーム「マンクスTT」は筐体が足を着かせないデザインになっていて好感がもてる。セガの最近のレースものは実在のモータースポーツに題材を求めたシミュレータ的作品ばかりで物足りなく感じないわけではないが，この際バイクの挙動がきちんとできていれば文句はない。プレイするまでは判断できないが楽しみ。 (A.T.)

▶かまたさんと久しぶりに会った。そのときにWindows用のDōGAシステムを見せてもらった。本格的なものではなく，「GENIE」のようなものだ。これが，結構よくできている。詳しくは来月号の連載で紹介する予定。あと，ぜんぜん関係ないが，今月は忙しいときに家の湯が出なくて風呂に入れないわ，万馬券3本買い損ねるわ，さんざんであった。 (高)

▶「ごきげんモコナ」がほしい。最近なぜか少し前から某電機店のレジの横にあるモコナのぬいぐるみが気になってしようがない。しかし，この歳でぬいぐるみを買うことに多少の抵抗が……ええい，なにをためらっているのだ。いますぐにでも秋葉原へ向かい，「ごきげんモコナ」をゲットするのだ！

(Oh!XさんちのJ君この頃少し変かも……のJ)

▶「アニメにはCD-ROMが大前提」ねぇ……。AMIGAが日本で流行らなかったのは不幸なことかもしれない。さて，PC-9801用のPC-FXボードが結構面白そうだ。ゲーム機本体の部分は置いといても，MotionJPEGとポリゴン描画チップがパソコン側で使えるってのはおいしい。こういうのが売れるようなら日本のパソコン界も安心なのだが？ (U)

▶かつてX68000シリーズに関わっていた人が各方面で活躍しているのは嬉しくもあり寂しくもある。もちろん，まだまだX68000シリーズを活用している人もいるわけで，はっきりいって脱帽ものだ。一方，最近になってはじめてX68000を手に入れたという初心者の方もいるようで，ちょっと心配だ。そういう人にもOh!Xは何かを残せるだろうか。 (T)

microOdyssey

最近，ゲームを遊ぶことに対する情熱というか執着心が薄れてきている。だからといって，ゲームを遊ぶことに飽きたわけではない。現実に，ゲームを遊ぶときには1日中遊びほうけていることもあるのだが(仕事は?)，1つのゲームを2，3日ほど遊ぶと放り出してしまい，すぐ次のゲームを探すことが多くなった。特にここ数年，コンシューマ機に手を出したときからその傾向が強まっている。

要因としては，手広くゲームを遊ぼうと欲張ってしまう状況が当たり前となってしまい，以前のように1本当たりにかける情熱が薄くなっていると考えられる。発売されるゲームの数が増えたため，面白そうなゲームを選んで遊ぶだけでもちょっとした数に昇る(もちろん，つまらなそうゲームだったら遊ばない)。ちょいちょいつまみ食い状態でも，割と時間が食われる状況に陥っているのだろう。

そう思いながら，もう少し突っ込んで，自分のゲームに対する姿勢というものを振り返ってみる。すると，実際にゲームに触れることなく雑誌などに掲載されている情報のみで，ゲームを批判しているときがある。別に遊ばなければ，批判してはいけないわけではないが，ゲームを見てくれだけで判断し，批判する。これは，なまじ手広く遊んで数をこなしたつもりでいるため，ついついうわべだけの知識でもって評価を下してしまう思い上がった部分が自分にあるのであろう。そうなると面白いと思ったゲームに対しても，すぐに底が見えたと勘違いして，次のゲームを探し始め，必然的にゲームのプレイも荒れてくるのだ。

こういう態度は，ゲームを制作する立場として，決してプラス方向に働くとは思えない。自分の中だけで完結している考えで，ものごとを図ることに慣れてしまうと，そこから生み出されるものもいわゆる自己中心的なひとりよがりな作品でしかないからだ。

それに，ゲームを遊ぶのはユーザーであり，受け手であるユーザーの立場としての気持ちを理解せずに作ったゲームが，評価されることはない。これははっきりしている。いままで，ユーザーを無視した制作者の心ないゲームが，どんな扱いを受けてきただろうか。そのほとんどが，ただの自己中心的な面白くないゲームとして批判され，闇に葬られてきた(あまりに突飛なゲームは，別の意味でユーザーの心に残ることがあるが)。

結局，ユーザーの立場である自分が，このようないい加減な態度でいると，制作者側に立ったときも，対象としようとしているユーザー像はいい加減なものしか思い浮かばないというわけだ。どうりで，ゲームを作っていても明確なターゲットが見えないはずだ。気づくのがちょっと遅かったかもしれないな。

ま，このようなことを思い返しても，いまさらどうしようもないといってしまえばそれまでだが，わずかながらに時間がある。即座に解決策が見い出せるものでもないが，直せるところは直して，できるかぎり努力するしかない。期待してくれている読者の方々を裏切らないためにも，許された時間を使って，せいぜいあがいてみるしかないだろう。　(J)

1995年12月号11月18日(土)発売

特別企画　MerryX'mas PRO-68K

・シューティングゲームSION IV+SLASH ver.3.0
・Z-MUSIC ver.3.0(お試し版?)
・SX-BASICコンパイラ 他

Oh!X 8周年特別企画(?)

試用レポート　16Mバイト増設メモリ「TS-6BE16」

特別付録 5"2HD(2枚組)　特別定価 1,280円(予価)

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 11月号
■1995年11月1日発行　定価760円(本体738円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印　刷　凸版印刷株式会社

満開の電子ちゃん

作・え 岡村

パソコンショップ満開の電話は03(3985)5530です。



第89号(9/18発送)には、オブジェクト指向を取り入れたLISPインタプリタとか、電子ブック辞書検索ソフトとか、AWESOME-Xレポート＆ツールなどで充実の秋。

講読方法：定期購読、ソフトベンダーTAKERU、NIFTY-SERVEでお買い求めいただけます。
また、JCB、VISAカードもご利用になれます(金額9,000円以上の場合)。
★定期購読(送料サービス、消費税込)3ヶ月＝4,500円、6ヶ月＝9,000円、12ヶ月＝18,000円。
・現金書留：〒171 東京都豊島区西池袋5-17-11 ルート西池袋ビル901 ㈱満開製作所
・郵便振替：02810-13298 口座名 電脳倶楽部
・JCB・VISAカード：フリーダイヤル0120-887780または、NIFTY- SERVE GO MANKAI。
ご注文の際には、郵便番号、住所、氏名、電話番号、タイプ(5インチ・3.5インチ)新規購読か継続購読かを必ずお知らせ下さい。新規購読の際、購読開始号のご指定のない場合は既刊の最新号よりお送りいたします。製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しいたします。
★TAKERUでお求めの場合、75号までは1,200円(税込)、76号以降1部1,600円(税込)です。
★お問合わせ先 TEL03-3985-6110(月～金 午前11時～午後6時)。
★バックナンバーは創刊号よりございます。★フリーダイヤルは、月～金 午前10時～午後5時。



(福岡県) 大塚 俊治

どうも皆さんこんにちは。電脳倶楽部、読んでますか？ ためになる各種講座、かしこくなれるPDD、イカすプログラム講座、華麗な電脳画廊、そして、変倉長のありがたいお言葉。皆さんはどれがお気に入りでしょうか。私のいちばんのお勧めは、なんといっても〝紳士と淑女の社交場〟です。その名にふさわしい話題から、ふさわしくない話題まで盛り沢山。とても楽しいです。私もときどきお邪魔してます。
それにしても、今月の満開の電子ちゃんも面白いですね。
それでは皆さんさようなら。

MPUアクセラレータ
（東京システムリサーチ）

⦿Xellent30（XVI用）
定価￥59,800⇨特価 **￥45,000**

⦿Xellent30s（ACE、EXPERT(II)、SUPER用）
定価￥54,800⇨特価 **￥41,000**

（●MPU交換に付き、保証(メーカー、当社)は付きませんので、ご承知下さい。）

P&Aならではの 新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証（メーカー保証1年＋P&A保証4年）
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
②中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証!!）
③初期不良交換OK!!（※新品商品に限らせていただきます。）
④永久買取保証
⑤配達日の指定OK!!（土曜・日曜・祭日もOK!!）
⑥夜間配達もOK!!（※PM6:00～PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

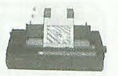
①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利!!
③月々の支払いは￥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括支払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になり
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

周辺機器コーナー （送料￥1,000・消費税別）



カラーイメージスキャナ（ケーブル付）
■JX-330X（SHARP）
特価**￥86,800**
■GT-6500WINS（エプソン）
特価**￥35,800**

カラーイメージジェット 限定5台



■IO-735X-B
定価￥248,000
特価**￥89,000**

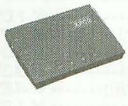
ビデオスキャナー
■CZ-6VS1
定価￥178,000
特価**￥129,000**



FDD（5インチ×2基）
■CZ-6FD5
定価￥99,800
P&A超特価
￥49,800

プリンター（ケーブル付）
- MJ-800C （エプソン）…特価￥60,300
- MJ-500C （エプソン）…特価￥40,300
- MJ-900C （エプソン）…特価￥81,300
- MJ-5000C （エプソン）特価￥135,300
- BJC-400J （キャノン）…特価￥38,800
- BJC-600J （キャノン）…特価￥51,300
- BJC-35V （キャノン）…特価￥40,800
- BJ-30V （キャノン）…特価￥30,300

ビデオ・スキャンコンバータ・ユニット
■XPC-2



定価￥39,800
▶特価**￥31,000**

- CZ-6BV1………定価￥21,000▶特価￥15,900
- CZ-8NM3………定価￥ 9,800▶特価￥ 7,200
- SH-6BF1………定価￥49,800▶特価￥36,500
- CZ-6BS1………定価￥29,800▶特価￥21,500
- CZ-8NJ2(限定)…定価￥23,800▶特価￥13,800
- CZ-6CS1(674C用)・定価￥12,000▶特価￥ 8,900
- CZ-6CR1(RGBケーブル)・定価￥ 4,500▶特価￥ 3,600
- CZ6CT1(テレビコントロール)・定価￥ 5,500▶特価￥ 4,400
- CZ-5MP1(X68030用)・定価￥54,800▶特価￥42,000
- TN-800TVEM（ビデオスキャンコンバータ・東京ニーズ）……………………特価￥27,800

送料￥700・消費税別

■システムサコムボード
- SX-68MII（MIDI）
定価￥19,800
特価￥13,500
- SX-68SC（SCSI）
定価￥26,800
特価￥17,500

X68000用ソフトコーナー （送料￥700・消費税別）

〈シャープ〉
- MUSIC PRO68K(MIDI)(CZ-247MSD) ……………………特価￥20,500
- CANVAS PRO68K(CZ-249GSD)特価￥22,000
- Easypaint SX-68K(CZ-263GWD) ……………………特価￥ 9,800
- Easy draw SX-68K(CZ-264GWD)・特価￥15,300
- New Print Shop Ver.2.0(CZ-265HSD) ……………………特価￥15,400
- Press Conductor PRO68K(CZ-266BSD) ……………………特価￥22,000
- CHART PRO68K(CZ-267BSD)・特価￥29,800
- EG-Word(CZ-271BWD)………特価￥44,900
- Communication SX68K(CZ-272CWD) ……………………特価￥14,500
- Datacalc SX-68K(CZ-273BWD) ……………………特価￥44,000
- MUSIC SX68K(CZ-274MWD)…特価￥29,300
- SOUND SX68K(CZ-275MWD)…特価￥11,500
- フォント・アンド・ロゴデザインツール SX-68K (CZ-282BWD)………………特価￥22,000
- BUSINESS PRO68K(CZ-286BSD) ……………………特価￥20,500
- SX-WINDOWディスクアクセサリー集(CZ-290TWD) ……………………特価￥11,500
- XDTP-SX68K(CZ-291BWD)……特価￥26,900
- C-Compiler PRO68K Ver.2.1(CZ-295LSD) NEW KIT……………………特価￥32,500
- SX-WINDOWS Ver.3.1(CZ-296SS/SSC) ……………………特価￥17,600

〈計測技研〉
- Double Bookin……………特価￥ 9,600
- CD-ROM Driver V.2.0………特価￥ 3,800
- シャーペン・ワープロパックVer2.0…特価￥ 7,800

〈その他〉
- F-Card V5 for X68K（クレスト） ……………………特価￥ 9,600
- F-Calc for X68K（クレスト）……特価￥11,000
- たーみのる2(SPS)……………特価￥13,000
- MU-1GS(サンワード)…………特価￥21,000
- マチエール V2.1(サンワード) ……………………特価￥28,800
- Z's STAFF PRO68K Ver.3.0(ツァイト) ……………………特価￥37,500
- Z's TRIPHONYデジタルクラフト(ツァイト) ……………………特価￥27,000
- XL/Image(IMAGICAテクノシステム) ……………………特価￥46,000

〈ゲーム〉在庫限り
- 魔法大作戦(X68/5")…………特価￥ 7,300
- パックランド(X68/5")…………特価￥ 6,200
- 餓狼伝説(X68/5")……………特価￥ 6,600
- スーパーストリートファイターII(X68.5") ……………………特価￥ 7,300

P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00～PM7：00 日・祭：AM10：00～PM6：00
☎03-3651-0148(代)
●定休日/毎週水曜日
FAX.03-3651-0141 MAC/DOS Vフロア ☎03-3655-4454

全国通販

★頭金なし!
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK!TELください。

P&A特選 今月の中古特選品



単品
●CZ-500CB
￥145,000



●CZ-623C
●68000専用モニター付
￥89,000



●CZ-653C
●68000専用モニター付
￥69,000

新品限定
- CZ-652C ……￥35,800
- CZ-653C ……￥37,800
- CZ-663C ……￥39,800

- CZ-600C‥￥30,000
- CZ-601C‥￥30,000
- CZ-611C‥￥32,000
- CZ-652C‥￥29,800
- CZ-612C‥￥50,000
- CZ-603C‥￥45,000
- CZ-653C‥￥30,000

- CZ-612C‥￥55,000
- CZ-623C‥￥65,000
- CZ-674C‥￥49,800
- CZ-634C‥￥98,000
- CZ-644C‥￥110,000

※上記は単品価格、モニター別売。

高額買取り（新品もOK）格安販売

■まずはお電話下さい。
下取り専用
買取り電話▶ ☎03-3651-1884 FAX.03-3651-0141

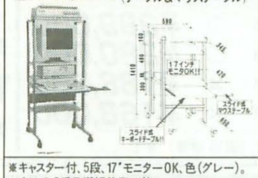
買取り価格…完動品・箱／マニュアル／付属品の価格です。中古販売…1年間保証付。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

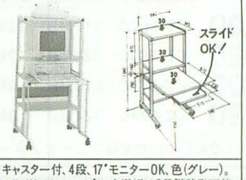
P&Aオリジナル特選 パソコンラック&OAチェアー （消費税込み）（送料無料、離島を除く）

①￥17,304（スライド式・キーボートテーブル&マウステーブル）



※キャスター付、5段、17"モニターOK、色(グレー)。
※上から2番目棚板移動可能。

②￥12,360（マウステーブルスライドOK）



※キャスター付、4段、17"モニターOK、色(グレー)。
※スライドマウステーブル、中棚板は2段階移動可能。

①￥4,944



- 布張り 色(グレー)
- ガス圧シリンダー

②￥6,283



- 肘付
- 布張り 色(グレー)
- ガス圧シリンダー

※ラック、チェアー持ち帰り可能です。ご来店下さい。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕 さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー



超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.6	3.0	4.2	4.89	6.5	10.0	14.3	18.9	24.3	31.8

（※車でお越しの場合は北海道拓殖BK前の新小岩駐車場をご利用下さい。）

※お支払いは、便利な商品到着払い〈手数料(10万円まで900円)要〉をご利用下さい。



ツクモイメージキャラクター
生田智子

X68。キーワードはツクモにあるゾ!!

MO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO

お申し込みは今すぐ！
受注専門フリーダイヤル
0120-377-999

◆本体・ディスプレイ・プリンタ・周辺……オリジナルと豊富な品揃!!◆

本体

CZ-674C-H(X68000 CompactXVI)



TS-XFDCAを使えば、
縦置き5インチモデル
X68000ｼﾘｰｽﾞ(PROｼﾘｰｽﾞを除く)
を外部ﾄﾞﾗｲﾌﾞとして使用可能！

是非、2台目のマシンとしてどうぞ！

※モニタ別売です

超特価 ¥78,000

お勧めのセット1

CZ-674C-H・・・・・ オープンプライス
PC-TM151(NECディスプレイ・ケーブル直付) ¥79,800



＊写真とセットのモニタは異ります

特価 ¥137,000

X68030 お勧めのセット2

CZ-500C-B・・・・・・ ¥398,000
外付500MBハードディスク サービス



特価 ¥228,000 ※モニタ別売です

満開製作所の商品も取扱中！

X68000 CompactXVI 24MHz改

RED ZONE ・・・・・・・・・ 特価 ¥ 98,000
RED ZONE(2DD) ・・・・ 特価 ¥ 103,000

満開製外付け5インチFDD

MK-FD1 ・・・・・・・・・・・・ 特価 ¥ 39,800

SCSI2 (FAST SCSI)カード

Mach-2 ・・・・・・・・・・・・・・ 特価 ¥ 28,800

X680x0シリーズ用RAMボード

SH-6BE1-1ME	(CZ-600C専用)	¥10,500
PIO-6BE1-AE	(ACE/PRO/PRO2シリーズ用)	¥10,500
PIO-6BE2-2ME	(拡張スロット用)	¥19,800
PIO-6BE4-4ME	(拡張スロット用)	¥33,800
SH-5BE4-8M	(X68030シリーズ用)	¥42,800
X SIMM VI	(XVI専用)	¥13,200
X SIMM VIc	(CompactXVI専用)	¥13,200
X SIMM10+8MBSIMM	(拡張スロット用8MB)	¥51,800
X SIMM10+10MBSIMM	(拡張スロット用10MB)	¥62,800

★当社でお取り扱いの商品は、お客様による改造機での動作保証は、一切致しません。

XsimmVI/VIc/TS-6BS1mkII用
8MB72Pin60nsパリティ無しSIMM
特価 ¥35,000
★各SIMMマザーカードとセットの場合
特価 ¥33,000

MPUアクセラレーターカード

お待たせしました!X68000PROシリーズに030パワー新登場!!
MC68030環境＋αがお手ごろ価格です！

MC68000モードとMC68030モードをソフトウェアにて切り替え可能ですので、既にお手持ちのソフトが動作しなくなる心配はありません。取付はドライバー1本でOKです。通常の動作速度向上はもちろん！レンダリング等の高精度演算処理に威力を発揮するMC68030モード用コプロセッサを登載しておりMPUからダイレクトに制御する専用プログラムがあれば、さらに動作速度が向上します。

新製品　CZ-652/662/653/663専用　T.S.R製 **Xellent30PRO** 定価¥54,800　特価 ¥41,800

売れてます!!　CZ-601/611/602/612/603/613/604/623専用　T.S.R製 **Xellent30s** 定価¥54,800　特価 ¥41,800

大好評発売中　CZ-634/644専用　T.S.R製 **Xellent30** 定価¥59,800　特価 ¥45,800

取付費別（店頭持ち込み時¥5,000、7日程度の日数を頂きます）※Human Ver3.0以外のOSは1995／9／25 現在対応しておりません。

DSPプロセッサカード

可能性は無限大!!DSPを操り高速演算、EIAJ光デジタル入力で高品質音声録音ができる！　また、別売り赤外線I/Fで、リモコン制御、電子手帳データー交換……なども。

GRAVIS製
AWESOME-X
定価¥89,800
特価 ¥79,800

マウス延長ケーブル(1.5m) **TS-MEXCB**・・・・特価 ¥1,880

X680x0ユーザーの為の ツクモ オリジナル シリーズ

X68000Compact/RED ZONE用 内蔵6MB+FPUボード

TS-6BE6DP



※FPUにMC68882を使用しているため、HumanVer3.0より前に付属していたFLOAT3.Xでは使用できませんのでご注意ください。
★大好評につき、若干納期を頂く場合がございます。ご了承下さい。
定価¥64,800
特価 ¥57,800

キーボード延長ケーブル(1.5m)

TS-KEXCB
特価 ¥1,880

ジョイスティックパラレルインターフェイス

●拡張スロットを使用しません。ジョイスティック端子に接続できるパラレルインターフェイスです。これでスキャナも高速で取り込みが可能になります。★取り込みソフトェア及びサンプルソース付属。

TS-JPIFE (EPSONｽｷｬﾅ対応用)



定価¥17,800
特価 ¥14,800

Matier Ver.2.1 対応！

TS-JPIFS (CZ-8NS1対応用)



定価¥17,800
特価 ¥14,800

ツクモオリジナルX680x0 HG

		本体	HDD	RAM	コプロ	特価
X68030	HG500	CZ-500C	515MB	12MB	○	¥299,000
	HG320	CZ-500C	324MB	12MB	○	¥280,000
X68000	HG500	CZ-674C	515MB	8MB	×	¥188,000
	HG320	CZ-674C	324MB	8MB	×	¥168,000

★HGシリーズのお問い合わせはニューセンター店（担当 伊藤）まで

SCSI+SIMMマザーカード

TS-6BS1mkII'
★X68000PROシリーズにも対応
★SCSIROMをさらにチューンアップ
もう少しだけお待ちください！
予価 ¥39,800 近日発売

TS-6BS1mkII用チューンROM
もう少しだけお待ちください！
予価 ¥9,800 近日発売

16MB増設メモリ

TS-6BE16(Xe30) Xellent30用
TS-6BE16(CZ50) CZ-500C/510C用



定価¥89,800
特価 ¥79,800
※フラットイメージでの増設ではありません。
このメモリボードへのアクセスには対応ソフトが必要です。

ディスプレイ

CZ-608D (14型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲ) ……特価 ¥66,000
CZ-615D (15型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲﾃﾚﾋﾞ) ……特価 ¥132,000
CZ-621D (21型ｶﾗｰﾃﾞｨｽﾌﾟﾚｲ) ……特価 ¥125,000

プリンター

ケーブル別売 セット特価¥3,000!!ただREDZONE用は¥5,500

●Canon
BJC-35v ……特価 ¥42,800
BJC-400J ……特価 ¥36,800
BJC-600J ……特価 ¥49,800

●EPSON
MJ-500C ……特価 ¥37,800
MJ-800C ……特価 ¥59,000
MJ-900C ……特価 ¥79,800

スキャナ

SHARP **JX-330X** ●SCSI接続 対ハーフケーブル付 特価 ¥89,800
SHARP **CZ-8NS1** ●接続ケーブル付 特価 ¥44,800
EPSON **GT-6500WINS** ●SCSI接続 ケーブル別 特価 ¥39,800

モデム

サン電子 MS144AF 特価 ¥12,800
MS288AF 特価 ¥26,800

ターミナルソフト

SPS たーみのる2 ……特価 ¥13,000
SHARP Communication SX-68K ＊要SX-Ver.3.0以降 ……特価 ¥15,800

HDD ★新製品続々登場中!お問い合わせ下さい!

I/Oデータ **HDS-540M** (H-Hケーブル付)…… (540MB) 特価 ¥26,800
I/Oデータ **HDS-1G** (H-Hケーブル付)…… (1GB) 特価 ¥55,800
KONIC **VIP-540CX** (H-Hケーブル付)…… (540MB) 特価 ¥29,800
KONIC **VIP-1080CX** (H-Hケーブル付)…… (1GB) 特価 ¥49,800

★VIPシリーズはブラックなので68にぴったり！

MO ★新製品続々登場中!お問い合わせ下さい!

Logitec(ケーブル・メディア別)
LMO-400 …… (230MB) 特価 ¥58,800
Logitec(ケーブル・メディア別)
LMO-450H …… (230MB) 特価 ¥74,800
ELECOM(H-Hケーブル・メディア付)
EMO-2300S …… (230MB) 特価 ¥75,800

9月26日現在、メーカー未出荷のものに関しては、出荷時の仕様変更によりお手持ちの機種で動作しない場合もあります。お問い合わせ下さい。

【東　京】●パソコン本店（各種パソコン・周辺機器）●本店IIWindowsタワー・（パソコン・ワープロ）●DOS／Vパソコン本館（DOS／Vパソコン・Mac・下取り）●万世店（総合通信機器）●5号店（ビデオ・ムービー・CS）●ソフト8号店（ゲーム機・ゲーム用ソフト）●買取センター（ゲーム機・ゲーム機用ソフト買取り）●ニューセンター店(各種パソコン・中古・下取り・買取り)　【名古屋】●名古屋1号店(パソコン全般)●名古屋2号店（パソコン全般・総合通信機器・ビデオ）　【札　幌】●札幌店（パソコン全般・総合通信機器）●DEPOツクモ札幌（パソコン全般）

X68とCD-ROMがもっと密に…

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO

受付時間 (平日) AM10:45～PM7:30
(日・祝) AM10:00～PM7:00
通販の(休) 第3木曜日

『FAX24時間お見積もり受付』
03-3255-4199
お名前.住所.電話番号.FAX番号をご記入の上ご依頼下さい。



ツクモグローバルJCBカード
JCBならではの国内・海外サービスにツクモオリジナルの特典をプラス。ツクモ各店にある入会申込書にてお申し込み下さい。くわしくはグローバル事務局03(3251)9898又は各店へ。
※ジャックス・VISA・セントラル・マスターも取り扱っております。

映像関連周辺機器

動画を始めてみませんか？

ビデオ入力ユニット **CZ-6VS1**
定価￥178,000
MC68EC020(25MHz)の32BitMPUを搭載し、SCSIを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。但しX680x0シリーズでご使用の場合には6万5千色までの表示となります。



特価 ￥135,000

多機能対応型スキャンコンバーター

電波新聞社 **XVGA-1V**
定価￥66,800



特価 ￥56,700
X680x0シリーズやその他のパソコンの水平周波数(24KHz/31KHz)をNTSC標準信号に変換するスキャンコンバータユニットですので、家庭用テレビやビデオ・デッキで映像を表示または録画することができます。また、ビデオプリンターを使えば画面のハードコピーも可能です。

XVGA OVERLAY UNIT
定価￥45,800
「XVGA-1V」に接続して、パソコンとビデオの映像を合成する拡張機器です。
X68でコントロールできる！(RS-232C接続)



特価 ￥38,900

XAV-2s
定価￥10,000



パソコンやゲーム機から出力される水平同期周波数15kHzのアナログRGBの映像信号を家庭用テレビで見られる映像信号(ビデオ/Sビデオ)に変換する装置です。
特価 ￥8,500

CD-ROM

大容量で大量頒布に適したメディア「CD-ROM」。ここではX680x0への導入方法を紹介します!!

4倍速

メルコ **CDS-4E** 特価 ￥23,800
Panasonic **LK-RC504NZ** 特価 ￥26,800
Logitec **SCD-430** 特価 ￥23,800
I/Oデータ **CDG-TX4A** 特価 ￥23,800

倍速7連装

ナカミチ **MBR-7.4PC** 特価 ￥29,800
MBR-7PC 特価 ￥19,800

ISO-9660フォーマットのCD-ROMを読みたい
計測技研製
◆「CD-ROMドライバーVer.2.1」◆
定価￥4,800 特価 ￥4,320
◆「シャーペンワープロパックVer.2.0」◆
★機能限定版CD-ROMドライバ付属
定価￥9,800 特価 ￥8,820
※SCSI2対応のドライブで利用可能ですが、当店では上記以外のドライブの動作確認はしておりません。

SX-WINDOW上でPhoto-CDを表示したい
計測技研製
「SX-PhotoGallary」
定価￥15,800
本店4F設置の"タケル"にて好評発売中!!
※Photo-CDポートフォリオには対応しておりません。

SX-WINDOW上でEPWING規約CD-ROMを活用したい
計測技研製
◆「SX-広辞苑」◆
定価￥19,800 特価 ￥17,800
複合要素検索 「広辞苑.X」
単要素検索 「LightWing」
◆「シャーペンワープロパックVer.2.0」◆
★LightWingのみ付属
定価￥19,800 特価 ￥8,820

X680x0シリーズ専用 CD-ROM
◆ソフトバンク社刊◆ 『SX-WINDOW開発キットVer.3.1』‥‥定価￥5,631
※書籍扱となってます 『NetBSD/X68k』‥‥‥‥‥‥‥‥‥‥近日発売予定！

MIDIコンピュータ特選セット

SC-55mkIIセット
SC-55mkII ‥‥￥55,000
SX-68MII ‥‥‥￥19,800
特価 ￥57,800

SC-88VLセット
SC-88VL ‥‥‥￥69,000
SX-68MII ‥‥‥￥19,800
MIDIケーブル(1m) ‥‥￥1,000
特価 ￥69,800

コードレスタブレット

高機能はそのままで、お手頃価格で再登場！爆発的な人気のお絵描きアイテム!!
NSCalcomp **DrawingSlateII**
定価￥34,800

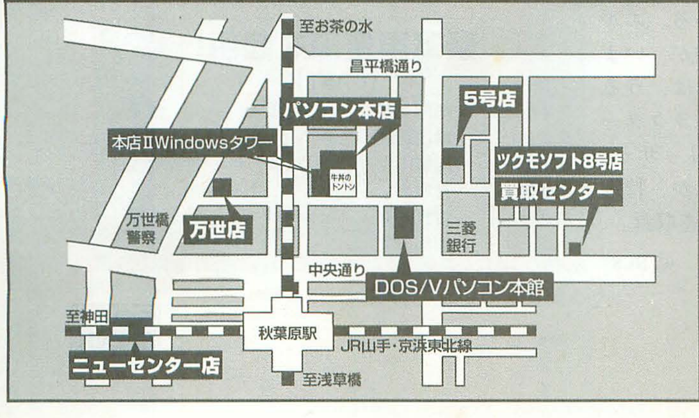


特価 ￥31,800
※付属の変換アダプターを使用する場合は、背面に20cm程の余裕が必要です。スペースに余裕が無いという方には、15cm程度の空きで使用可能な変換ケーブルをご用意しております！お気軽にお問い合わせ下さい。
‥‥‥‥‥‥‥‥‥‥特価 ￥1,800

ソフトウェア 特価

SX-WINDOW Ver3.1システムキット	￥18,200
SX-WINDOWデスクアクセサリ集	￥11,800
Easydraw SX-68K	￥15,800
Easypaint SX-68K	￥10,200
Communication SX-68K	￥15,800
SX広辞苑(CD-ROM別)	￥17,800
EGWord SX-68K	￥47,800
SX-WINDOW開発キット	￥31,800
MUSIC SX-68K	￥30,400
XDTP SX-68K	￥28,000
DataCalc SX-68K	￥47,800
フォント&ロゴデザインツール書家万流SX-68k	￥23,800
C COMPILER Ver2.1 NEWKIT	￥35,800
Matier Ver2.1	￥29,800
XL/Image	￥49,300
CD-ROM Driver	￥4,320

秋葉原



名古屋



札幌



お支払い

あなたのご都合に合わせて…

クレジット
月々￥3,000以上の均等払いも頭金なし。夏・冬ボーナス2回払いもOK！

カード払い
￥5,000以上
通信販売での御利用カード
ツクモグローバルカード・セントラル・ジャックス
※御本人様より電話で通信販売部へお申し込み下さい。

各種リース払い
詳しくは各店にご相談下さい。

現金書留払い
〒101-91 東京都千代田区神田郵便局私書箱135号
ツクモ通販センター Oh!X係

代金引き換え配達
お申し込みは電話1本でOK！
配達日の指定もできます。

銀行振込払い
事前にTELでお届け先をご連絡下さい。
三和銀行 秋葉原支店
(普) 1009939 ツクモデンキ
※振込手数料はお客様の負担となりますのでご了承下さい。

商品についてのお問い合わせは各店に

秋葉原

(営) 平日AM10:45～PM7:30・日祝AM10:00～PM7:00
(休) 第3木曜日

ツクモパソコン本店 4F
03-3253-1899
03-3253-5599(代)

ツクモニューセンター店
03-3251-0987

名古屋

(営)平日AM10:30～PM7:30 土・日・祝AM10:00～PM7:30

ツクモ名古屋1号店
052-263-1655
第1アメ横ビル内 (休)火曜日

ツクモ名古屋2号店
052-251-3399
第2アメ横ビル内 (休)水曜日

札幌

(営)平日AM10:45～PM7:30日・祝AM10:15～PM7:00

●両店ともX68関連商品はお取り寄せのみ(展示等はありません)となります。ご了承下さい。

ツクモ札幌店
011-241-2299 (休)木曜日

DEPOツクモ2番街店
011-242-3199 (休)木曜日

★商品はお電話受け付けより、標準日数3日～1週間でお届け致します。(一部地域を除く)
★表示価格には消費税は含まれておりません。

安いのに親切
TSUKUMO

九十九電機株式会社

SOFTBANK GAME BOOK SELECTION

SOFTBANK CD・BOOK

メモリアルドラマCD＆ファンブック

ファンタシースター



好評発売中!

セガのゲームマシン（マークIII、メガドライブ）を代表するファンタジーRPG「ファンタシースター」ファンブック。セガ監修によるオリジナルドラマCD付き。ドラマCDは、シリーズ4作目「千年紀の終りに」の世界をベースに、シリーズで人気の高いキャラクター、ネイを使ったオリジナルシナリオによるものです。出演している声優のインタビューなども収録。

B5判・定価3,900円

人気声優が出演!

ネイ……………三石琴乃
スレイ…………井上和彦
フォーレン……速水奨
ルディ…………阪口大助

© SEGA

ファンブックと揃えて、
コレクターズアイテムに加えよう!

ファンタシースター 公式設定資料集

綴じ込み付録
米田仁士氏
描き下ろし
B4判
特製ポスター

「ファンタシースター」の公式設定資料集。これまでに発売されたIから「千年紀の終りに」までの未公開資料を含め、すべての資料を網羅。アルゴル太陽系年表完全版や用語集、開発者スペシャルインタビューなどから、ファンタシースター通販グッズ、すべてのテレビCMまで大紹介。

B5判・定価1,600円

10月下旬発売予定!



SOFTBANK CD・BOOK

メモリアルドラマCD&ファンブック

ラングリッサーII

好評発売中!

秘剣“ラングリッサー”をめぐって、激しい戦いの幕が開こうとしている。エルウィンとレオンとの宿命の戦いが、いま始まる!!そしてファンブックでは、うるし原智志氏デザインによるキャラクターの魅力を徹底紹介＆「ラングリッサー」シリーズの歴史を検証。そのほか、特別インタビューやメイキングなどを収録。

B5判・定価3,800円　©NCS

豪華声優陣を起用!

【CAST】

エルウィン …草尾　毅
シェリー ……横山智佐
リアナ ………國府田マリ子
エリザ ………林原めぐみ
エグベルト …青野　武
ナレーター …銀河万丈
ジェシカ……藩　恵子
ヘイン………山口勝平
レオン………置鮎龍太郎
レアード……堀川　亮
バルガス……郷里大輔

SOFT BANK

肩すかし、嵐の後の、秋の空。ジャストのX68kペリフェラル

・・今回は結構期待してたんですけどねぇ。小笠原の東側だし、気圧もムチャクチャ低いし、目もハッキリ見える。あの新宿での変電所火災の夜を超える一大イベント、本物の台風クラブを想像していた担当としては、ちょっと期待はずれでしたね。直撃した伊豆諸島や不幸にも亡くなられた方には失礼ですが、やはり変な期待をしていた方も沢山いたんじゃないでしょうか。でも、雨は一杯降って水不足の不安もなくなったし、結果オーライって感じですね。よっかたよかた。不謹慎と思いつつも、次は春先の大雪in首都高速に焦点を絞り、スタッドレスタイヤの選択を始めようかと思っていましたが、・・・・やられましたよ。
で、ここでコマーシャルです。(笑)。

▽拡張SIMM メモリーボード*ER10S*

型番：ER10S0n（SIMM未実装）定価￥14,800；ER10SDn（4MByte SIMM 1枚実装済）定価￥39,800　対応機種：X680x0 全機種（定価はすべて税別）

□クロックスピード20MHzオーバーのRISCチップを載せたプリンターがはびこる世の中、クロックスピード10MHzのX68000、今さらながら速い処理速度とは言えなくなりました。□68000の10MHzもさることながら、このクロックスピードに合わせたメモリー周辺の設計も足を引っ張る要因となっています。これではMPUのクロックを上げてもその効果が充分に生かされないこととなってしまいます。□H.A.R.Pの設計段階で判明していたMPUの高速化に伴うバス等でのウェイトタイム増大。この無駄な時間をより有効に活用するためのアーキテクチャーがER10の顔です。□H.A.R.P側から見た場合、MPU内部の倍速化された演算処理はストレートにバスに反映されるものの、メモリーアクセスに際しては既存クロックのサイクルで動作するバスのタイミングにあわせた動作をしなければならず、結果として常にウェイトが入っているような状態となります。□ここでER10をバスに接続した場合、バス側で4クロックをワンサイクルとするメモリーアクセスに対し、倍速動作のMPUクロックのアドバンテージを生かし、バス側で1クロック短縮した形でアクセスを完了できるようにタイミングを取る設計としています。□さらに、高速タイプの入手が容易な72ピンタイプのSIMMを採用、さらに内部で使用するゲートICなども高速のものを採用し、全体的な信頼性と安全性の向上に努めています。□「H.A.R.Pでない人」(笑) にもメリットがあると思いますよ。ER10、いかがですか。

▽MPUアクセラレーター*H.A.R.P* for MC68000

型番:DCMA00D1 定価￥29,800　対応機種:X68000初代,ACE,EXPERT,SUPER

マシンは速くしたい、改造は自信がない、費用も押さえたい。三拍子そろったあなたの欲求、H.A.R.Pがまとめてお引き受けいたします。□既存のMPUと交換するだけであっという間に倍クロック動作、周辺回路とのタイミングはクロックアップ前の状態を保ったまま、電気的に負担をかけることなく手軽に高速化。ソフトウェア的な互換性をパッチ不要のまま高いレベルで実現しています。□さらに拡張メモリーボードER10Sと組み合わせられることにより、メモリーアクセスのボトルネックを改善、トータルで約50%（弊社測定値）のパフォーマンスアップが可能です。 H.A.R.Pの性能を確実に引き出すには両者を組み合わせて使うのがベストですょ。□手軽なインストレーションと優れたコストパフォーマンス、H.A.R.Pは常にあなたの強い味方ですょ。

▽拡張 I/O スロット*ESX68*

型番：ESX68L4 予価￥39,800 対応機種：X680x0 全機種

OS-9をはじめ、実はFA系での隠れた需要もあるX680x0、この辺の用途にご利用の皆様には特に拡張 I/O スロットの少なさが問題となっているかと思います。□そんな需要家の皆様、そして純粋にコンピューティングを楽しむユーザーの皆様、外部拡張 I/Oスロットはいかがでしょうか？□本体電源に連動する外部スロット専用電源を内臓し、X68k本体とのインターフェースカードは高速タイプのバッファを搭載。加えて3スロットが追加利用できます。□LAN,PIO,GPIB, 入れたいカードは何でもどうぞ。□結構シリアスな設計しました。ESX68、くどいようですが、よろしくどーぞ。

▽MPUアクセラレーター*H.A.R.P-FX*（H.A.R.P for MC68030）

型番：DCMA30F1　予価￥54,000
対応機種：X68030をはじめ、MC68030（PGAソケット）が採用されたコンピュータシステム（供給クロック25MHz以下）

□X68030をはじめPGAパッケージタイプ68030を採用するパーソナルコンピューター、ワークステーションのほとんどに適応可能なMC68030互換MPUアクセラレーター、H.A.R.P-FXですX68030への実装時には 25MHzのクロックを2倍、オンボード上のMC68030RC50へフルスペック50MHzクロックを供給し、さらにMPUオンチップのキャッシュメモリーがクロックスピードと相乗し優れたパフォーマンスを発揮してくます。もちろん、ソフトウェアの互換性を完全に維持、既存の環境で動作していたソフトウェアならまず問題なく実行可能でしょう。Pentiumの120MHzもいいですが、68030の深い味わいを放つH.A.R.P-FX、ひたすら我が道を突き進みます。ご期待ください。

（前半から続く）・・・・真夜中の中央道。山肌に見える神社の鳥居を象った電飾に見入っているスキでした。小生が運転するEF3-51型シビックシャトル56j（5速マニュアル）は、事もあろうに某山○県警のパトカー（もちろん白黒）を追い越していたのでした。・・・・。39kmオーバー、高速道特例で青キップで済んだものの、反則金35,000円＋3点。前歴2回相当のこの身体に、降って湧いた湧いた欠格1年（あ、取り消しで1年免許が取れないことです。念のため）の危機！。担当はこのピンチを無事切り抜けることができるのかぁ?。危うし広告担当！以下次号だぁ。

サポート
(有)エヌ・エム・アイ

開発・販売
(株)ジャスト
〒156　東京都世田谷区宮坂3-10-7 YMTビル3F
Phone.03-3706-9766　FAX.03-3706-9761　BBS.03-3706-7134

GAME BEST SELECTION
ゲームベストセレクションシリーズ

ソフトバンク株式会社
出版事業部

●定価は税込です　●お近くの書店でお求めください

米国「Codies賞」受賞！
超話題の純国産シミュレーションソフトを完全攻略!!

絶賛発売中



Tower［タワー］
公式パーフェクトガイド

◎グレードを上げるための数々の条件をクリアし、思い通りのビルを建築する様々なテクニックを徹底解説。
◎秘密の裏ワザ、コマンドなども完全紹介。
◎困ったときにすぐ役立つ〈INDEX〉付き。

山猫有限会社 著

昨年発売された中で最も優れたソフトに与えられる権威ある「Codies賞」を受賞した、大ヒット純国産シミュレーションゲーム「Tower」公式完全ガイド。最高グレードである〈Tower〉の称号をもらうまでの様々なテクニック、自分の好きなビルを建築するためのノウハウなど、「Tower」のすべてを徹底解説！

A5判・定価1,600円

The PlayStation BOOKS

プレイステーションの定番ソフトを完全ガイド!!

中ボス・最終ボスの使用法まで完全ガイド！

ZERO DIVIDE
［ゼロディバイド］
パーフェクトガイド

好評発売中！



A5判・定価880円

プレイステーション用ロボット対戦格闘ゲーム「ZERO DIVIDE」を徹底攻略！プレイヤーキャラクター8体の対CPU・対人攻略を中心に、ボスキャラ・隠しキャラ「NECO」の操作法、隠しシューティングゲーム「Tinyファランクス」の紹介まで網羅した完全攻略本。「ZERO DIVIDE」をより一層楽しむための要素を満載した、ユーザー必携の一冊。

© ZOOM

これさえあれば好きなクラブで優勝できる！

Jリーグサッカー プライムゴールEX パーフェクトガイド

好評発売中！



A5判・定価880円

Jリーグ公認のプレイステーション用サッカーゲーム「Jリーグサッカープライムゴール EX」の完全攻略ガイド。全7種類にも及ぶモードの解説、基本テクニック、全クラブ選手データ、フォーメーション解説、高等テクニックを駆使した必勝パターン紹介など盛りだくさんな内容。この本があれば自分の好きなクラブで優勝できること間違いなし！

©（株）ナムコ

好評発売中！



究極のエースパイロットをめざせ！
エースコンバット パーフェクトガイド
A5判・定価880円



アークザ ラッドを100%遊び尽くせる！
アーク ザ ラッド パーフェクトガイド
A5判・定価880円



本誌で人気の格闘ライター杉澤教授が完全攻略！
鉄拳 パーフェクトガイド
A5判・定価880円



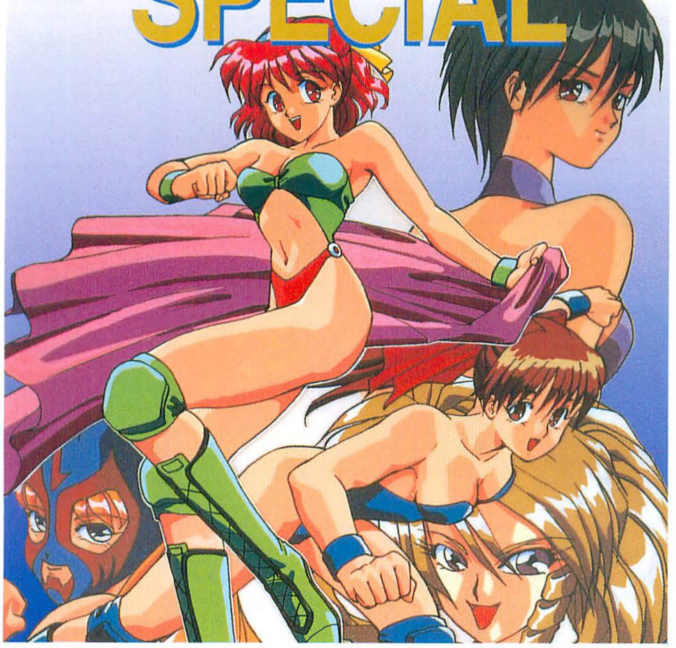
闘神伝の楽しさが、この一冊で倍増！
闘神伝 パーフェクトガイド
A5判・定価880円

SOFT BANK ソフトバンク株式会社/出版事業部
販売局 TEL. 03-5642-8101

●定価は税込みです ●お近くの書店でお求めください

ハイパーアクションプロレスカードゲーム

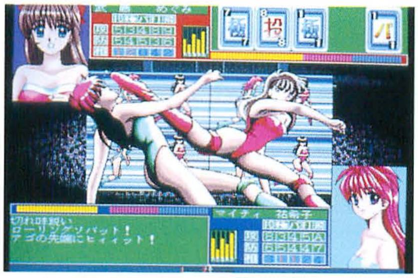
レッスルエンジェルス SPECIAL



セクシーでパワフルな女子プロを制覇しろ！

18禁版

カードバトルにプロレスを融合させた、「レッスルエンジェルス」シリーズ。いよいよ最大のヒット作「レッスルエンジェルススペシャル」が登場です。さまざまなイベントの選択によって運命が変わる、マルチシナリオ・マルチエンディング。プロレス技数、カテゴリーが増加して、レスラーの個性もパワーアップ。そして、「恐怖の水着はぎデスマッチ」もパワーアップして復活！18禁だから、そのセクシー度はもうケタ違い！待望のX68000移植完成！明日のトップイベンターを目指すのだ！











機能アップ！

- ●オリジナルオープニングを収録
- ●画面のレイアウトを変更
- ●エキジビションモードグラフィック描き直し
- ●256色モードと16色モードを搭載
- ●サウンドも明るめに変更
- ●AD-PCMによる効果音
- ●ディスクアクセスを最少に抑える設計

このソフトは、全国のパソコンショップで、パッケージ版で販売いたします。TAKERUでは販売致しません。TAKERU事務局では通信販売はいたしませんので、悪しからずご了承下さい。

対応機種：X68000/X68030
要メモリ2Mバイト
（ハードディスク対応）

制作：グレイト

¥8,800（税別）



三國志

知力の極限に挑む、君主、武将、軍師の膨大なデータ。小説よりリアルと、名作の誉れ高い中国統一ゲーム。この歴史的な傑作シリーズはどのようにして始まったのか?SLGファンなら絶対に見逃せない!!

制作/光栄
対応機種/X68000（30不可） **¥5,200**



太閤立志伝

裸一貫の足軽頭から身を興し、関白にまで登り詰めた男・木下藤吉郎(豊臣秀吉)。草履を温めたエピソード・奇跡の墨俣一夜城など、数々の逸話を持つ男の一生を再現する、リコエイションゲームの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**



ファランクス

デカキャラ・派手め演出の横スクロールオアフーシューティング。拡大・回転・縮小・多関節・半透明・ラスタースクロール・MIDIと、各種要素がいっぱい詰まってます。

制作／ズーム
対応機種／X68000（30不可） **¥2,500**



三國志 II

登場人物350余名、最大11人まで同時プレイ可能、6編のマルチシナリオ方式、埋伏の毒・駆虎呑狼等のユニークな計略要素導入、さらに深みを増した外交・HEX戦など、まさに名作!カシオペアの向谷　実のBGMも話題に。

制作/光栄
対応機種/X68000（30不可） **¥4,900**



蒼き狼と白き牝鹿　元朝秘史

光栄歴史三部作の一角を成す、草原の英雄チンギス・ハーン。稀代のスケールと空前絶後の迫力で、一代帝国を築き上げた男の豪快な一生を見事に再現!熱いシミュレーションの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**



A列車で行こうII

かの「A列車」シリーズの第２弾。パズル的要素がアツクなる!鉄道会社社長の立場で、線路の敷設・撤去を行い、ワールドワイドにマップを発展させていこう。

制作／アートディンク
対応機種／X68000（30不可） **¥3,800**



大航海時代

リコエイションゲームシリーズの傑作。毎回違った展開が楽しめるイベントジェネレーティングシステム。帆船の特徴が活かされたHEX戦。失われたロマンを求めて、冒険者たちの航海の旅が始まる。

制作/光栄
対応機種/X68000（30可） **¥3,400**



ロイヤルブラッド

新シリーズ「イマジネイションゲーム」のデビュー作。イシュメリアという架空の島国を舞台にした、幻想世界のシミュレーションゲームだ。あなたは独立貴族のひとりとなり、領主達が持っている6つの宝石を集め、イシュメリアの新王となれ!

制作/光栄
対応機種/X68000（30可） **¥2,700**



AIII（A列車で行こう3）

さらにワイドに、さらに完成度の増した、世界レベルヒットの第３弾。世にA.IIIブームを巻き起こしたことで、記憶に新しい超有名作、ついに文庫に登場！

制作／アートディンク
対応機種／X68000（30可） **¥3,800**



維新の嵐

坂本龍馬が、西郷隆盛が、吉田松蔭が日本を憂い、改革を目指して奮い立つ!幕末の志士の個性を際だたせる緻密なパラメータ。出会いの楽しさ、駆け引きを楽しむ新システム。強力な機能で、維新を操れ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**



ヨーロッパ戦線

戦乱のヨーロッパ。砂塵の彼方から迫り来る黒い車体は、敵か味方か?次々に飛び込んでくる情報、時事刻々と変わる戦局。多彩な兵器やユニット、人間的要素を重視した各種パラメータ。WWIIシリーズ第2弾。勝利の旗を手に入れろ!

制作/光栄
対応機種/X68000（30可） **¥4,500**



栄冠は君に

高校野球シミュレーションシリーズの、記念すべき第１作。全国制覇を達成するには、3990校の頂点に立たなければならない。感動の優勝セレモニーを、果たして見ることが出来るか！？

制作／アートディンク
対応機種／X68000 **¥3,800**



信長の野望　戦国群雄伝

400余名の群雄が割拠する下剋上の乱世。配下の羽柴秀吉、柴田勝家を個性豊かな武将たちを、思いのままに操って、戦雲たなびく戦場へ、天下分け目の決戦に臨む！光栄の代表作「信長の野望」シリーズの傑作！

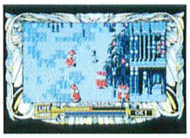
制作/光栄
対応機種/X68000（30可） **¥3,400**



大戦略 III '90

90年代にふさわしくパワーアップされた「大戦略」シリーズ。戦略思考ルーチン、ゲームスピード、コマンド体系、リアルタイムオペレーションなど大幅革新された作品です。

制作／システムソフト
対応機種／ X68000 **¥2,500**



ルーンワース「黒衣の貴公子」

ハイドライドシリーズに続く、新ARPGシリーズ第1弾。綿密に構築された世界「ルーンワース」を舞台に、極めて自由度の高いゲームシステムの中で、興奮の冒険が始まります。

制作／T&Eソフト
対応機種／X68000 **¥700**



伊忍道　打倒信長

1つのゲームでSLGとRPG、2つのジャンルが楽しめるリコエイションゲームの第3弾。特にRPGの要素が濃い、異色傑作だ!意志を持ったキャラクターが目的に向かって行動を展開。敵を倒して腕を上げ、技を磨いて信長を倒せ!

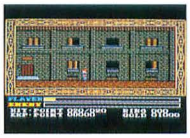
制作/光栄
対応機種/X68000（30不可） **¥3,400**



ジェノサイド 2

あのズームのゲームがついに名作文庫に登場！特大キャラとハデハデな演出で、68ユーザーのどぎもを抜いた名作アクションゲームだ。MIDIにも対応しているぞ。

制作/ズーム
対応機種/X68000（30不可） **¥2,500**



イース III（ワンダラーズフロムイース）

よりアクション性を増した、これまた、大人気を博したアクション・ロールプレイング。アドルの最後の冒険物語でした。攻撃方法もいっそう多彩になって、時間を感じさせない逸品です。

制作／日本ファルコム
対応機種／X68000（30不可） **¥2,000**

パソコンソフト
自動販売機
TAKERU

TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
プラザー技術開発センタービル2Ｆ
TEL(052)824-2493（受付時間：月〜金　13:00〜18:00）

営業所
東京営業所
(03) 5443-4967
大阪営業所
(06) 258-3024

通信販売 1994年4月1日より、送料／手数料が有料になりました。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。

T1002179110763 雑誌 02179-11